AN ABSTRACT OF THE THESIS OF

Yong Thye for the degree of Master of Science in
Electrical and Computer Engineering presented on
February 4, 1988

Title: COLAN IV, A Local Area Network For Communications and Control

Redacted for privacy

Abstract approved: ______________________
James H. Herzog

This thesis discusses the design and development of a viable local area network (LAN) which supports both communications and control applications. The emphasis was to develop a practical and cost effective system which offered high performance while utilizing low cost components. System cost was kept low by utilizing a twisted-pair cable as the communication medium. Economical and commercially available circuit boards were used to implement the network interface units.

The design of COLAN IV involves selection of network topology and media access technique. Bus topology was chosen for its simplicity and its capability to support decentralized control structure. The RS-485 bus standard has been adopted to achieve high performance with the twisted-pair cable. Designed to operate under light loads, COLAN IV uses the CSMA/CD media access method to optimize

channel throughput. Requiring no complex software algorithm nor complicated hardware, the CSMA/CD technique offers reliable operation, "fair" bus access, and fully distributed control structure.

The current configuration supports 63 network nodes and is expandable to 256 nodes. It employs a message switching technique for data communications among nodes at a data rate of 9600 baud.

Its hardware and software were developed to implement both communications and control applications. User applications implemented include distributed control system, data communications, electronic mail, and resource sharing. Designed as a dual purpose LAN, it can support several distributed control systems while operating as a communications network.

The features of COLAN IV have been tested with numerous investigative experiments performed with 4 nodes. Satisfactory results from these experiments confirm the viability of COLAN IV as a dual purpose LAN.

COLAN IV, A Local Area Network for Communications and Control

by

Yong Thye

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed February 4, 1988

Commencement June 1988

APPROVED:

Redacted for privacy

Associate Professor, Electrical and Computer Engineering in charge of major

Redacted for privacy

Head of Department of Electrical and Computer Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented February 4, 1988

Typed by Yong Thye for Yong Thye

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

COLAN IV, A LOCAL AREA NETWORK FOR COMMUNICATIONS AND CONTROL

CHAPTER 1

INTRODUCTION

1.1 Background

Although the computer industry is young compared to other industries (e.g., automobile and air transportation), computers have made spectacular progress in a short time. During the first two decades of their existence, computer systems were highly localized, usually within a single large room. The merging of computers and communications has had a profound influence on the way computer systems are organized. The old model of a single mainframe serving all of the organization's computational needs is rapidly being replaced by one in which a large number of autonomous computers are interconnected to form a computer network.

Through the 1970's, with the evolution of microprocessors, the concept of distributed processing became ever more attractive. The ability to integrate the intelligence of yesterday's boards onto today's chip drove the cost of computing to a point where decentralizing of processing was not only feasible, but practical as well.

In the 1980's the industry witnessed the advent of engineering workstations, instrumentation, automated equipment, and personal computers, all with the capability of performing their respective tasks independent of any supervisory machines. This decentralization of computing power has brought about the need to interconnect different "nodes" in order to allow the following:

1. Sharing of expensive resources like printers, plotters, and memory storage devices.
2. Sharing of information via common data bases.
3. Providing high level services, such as electronic mail and user applications.

Consequently the concept of computer networking was born.

The definition of a computer network, in its broadest sense, is a system of computing units interconnected by communication lines. This definition can be further broken down based on the characteristics of a specific network. One special type of network is a Local Area Network or LAN. As its name implies, a LAN is geographically localized. It is defined as a network supporting peer-to-peer communications over distances of tens of meters to several kilometers. Peer-to-peer implies that each station on the network is its own "boss." In other words, there is no master-slave relationship.

1.1.1 Evolution of Reference Model and Standards

The requirements of a LAN vary, depending on the environment. In general, there are three distinct environments: factory, engineering, and office. On the factory floor, where a typical application involves the interconnection of process control stations and robotics controllers, a network must be able to span large distances while maintaining a high degree of noise immunity. Furthermore, the ability for each node to send data must be deterministic, i.e., each must have a chance to transmit in a given interval of time. For engineering applications, in which CAD/CAE workstations are interconnected, the requirement is high throughput over short to moderate distances. Finally, in the office the primary need is the interconnection of personal computers and servers. The office requirement is highly cost sensitive. Because of the different applications and varied requirements, many types of incompatible LANs have evolved in the last decade. As a result, many LANs in industry are of a proprietary nature. An end user has to purchase all of the LAN computing components from the same vendor in order to set up a network.

The problem of incompatibility initiated the concept of "Open Systems." Open systems allow end users to purchase equipment from several vendors in order to realize an optimal solution for a given application. In an effort to encourage "open" networks, the International Standard

Organization (ISO) developed the Open Systems Interconnect (OSI) Reference Model.

The Institute of Electrical and Electronics Engineers (IEEE), in response to a need for standardization in the field of LAN, formed the IEEE 802 Standards Committee. The following sections describe the OSI Reference Model and the IEEE 802 standards.

1.1.2 The OSI Reference Model

In an effort to encourage "open" networks, ISO developed the OSI Reference Model for the design of computer networks. The OSI reference model logically groups the functions and sets of rules (protocols) necessary to establish and conduct communications between two or more parties. The model consists of seven sets of functions, often referred to as layers. The OSI reference model describes the functions of each layer in broad terms, not specific implementations.

This layered model approach affords two key advantages. First, layers allow a clear division of design tasks through modularity. Second, systems based on a layered architecture are flexible. Flexibility is achieved because each layer functions independently of the layer above or below it. Thus, specific layer implementations can be changed easily. The purpose of each layer is to offer certain services to the higher layers thus shielding them from the details of how the offered services are actually performed. Between each pair of adjacent layers there is an interface. The interface defines the types of primitive operations and services which are offered by the lower layer to the upper one. The seven layers of OSI model are summarized as follows:

Layer Name	Functions
Layer 1 (Physical Layer)	Transmits or receives a bit stream from the medium.
Layer 2 (Data Link Layer)	Transfers units of information from one node to another node.
Layer 3 (Network Layer)	Routes information.
Layer 4 (Transport Layer)	Provides end-to-end data integrity and quality of service.
Layer 5 (Session Layer)	Coordinates interaction among end-application processes.
Layer 6 (Presentation Layer)	Provides code conversion and data reformatting.
Layer 7 (Application Layer)	Provides appropriate services for applications.

Layer 1, the physical layer, specifies the type of connectors, signal levels, data rate, data encoding method, modulation method, and the procedures to establish, maintain, and terminate physical connections.

Layer 2, the data link layer, describes the rules for transmitting on the communications channel. It specifies the format of information, and the procedures for gaining the control of the channel.

Layer 3, the network layer, controls switching between links in a multihop network. The network layer is not necessary for a single, stand-alone LAN in which all stations share the same channel. This layer is critical in gateway, communication server, and dial-up communication applications.

Layer 4, the transport layer, ensures end-to-end message integrity and provides the required quality of service for information exchange. For example, end-to-end acknowledgements and flow control are performed by the transport layer.

Layer 5, the session layer, establishes and terminates logical connections between network entities. This layer is also responsible for the mapping of logical names into network addresses.

Layer 6, the presentation layer, provides any necessary translation, format conversion, or code conversion to put the information into a recognizable form.

Layer 7, the application layer, provides network based services to the end users. Distributed data bases and electronic mail are two examples of network services.

1.1.3 Implementation of the OSI Reference Model in a LAN

The first two layers of OSI Reference Model ensure interconnectivity of the "nodes" of a LAN. By implementing these two layers according to standardized specifications, equipment from multiple vendors can be physically and electrically interconnected. The remaining five layers of OSI Reference Model ensure operation among the interconnected stations in an open network.

1.1.4 IEEE 802 Standards for LAN

The ISO open systems interconnect model has been adopted by the IEEE Standards Committee to define the specifications for the data link layer and the physical layer. In response to a need for standardization, the IEEE 802 Standards Committee drafted the specifications of a LAN based on widely accepted industry standards. The IEEE 802 Standards classify LANs according to their network topologies and media access methods.

Standards	Media Access Method	Network Topology
IEEE 802.3	CSMA/CD	Bus
IEEE 802.4	Token Bus	Bus
IEEE 802.5	Token Ring	Ring

The IEEE 802 Standards for LANs consists of five parts: 802.1, 802.2, 802.3, 802.4, and 802.5. As depicted in Figure 1.1, IEEE 802.1 describes how each part of the 802

Standards fits into the OSI Reference Model. The IEEE 802.2 standard describes the functions and protocols of the logical link control sublayer in the LAN.

1.1.5 IEEE 802.3 Standard (CSMA/CD)

The Carrier Sense Multiple Access with Collision Detection (CSMA/CD) media access method is the means by which two or more stations share a common bus transmission medium. To transmit, a station waits (defers) for a quiet period on the medium (that is, no other station is transmitting) and then sends the intended message in bit-serial form. If after initiating a transmission, the message collides with that of another station, then each transmitting station intentionally sends a few additional bytes to ensure propagation of the collision throughout the system. The station remains silent for a random amount of time (backoff) before attempting to transmit again [IEEE 85a]. Each aspect of this media access method is specified in details in IEEE 802.3 Standards.

1.1.6 IEEE 802.4 Standard (Token Bus)

The essence of the token bus access method can be summarized as:

1. A Token controls the access to the physical medium; the station which holds the token has momentary control over the medium.
2. The token is passed by stations residing on the medium.

As the token is passed from station to station a logical ring is formed.

3. Steady state operation consists of a data transfer phase and a token transfer phase.
4. Logical ring maintenance functions within the stations provide for ring initialization, lost token recovery, new station addition to the logical ring, and general housekeeping of the logical ring. The ring maintenance functions are replicated among all the token using stations on the network.

Shared media generally can be categorized into two major types, broadcast and sequential. The IEEE 802.4 standard deals exclusively with the broadcast type. On the broadcast medium, every station may receive all signals transmitted. Media of the broadcast type are usually configured as a physical bus [IEEE 85b]. During normal, steady state operation, the right to access the medium passes from station to station in the sequential order of the logical ring.

1.1.7 IEEE 802.5 Standard (Token Ring)

A token ring consists of a set of stations serially connected by a transmission medium which forms a ring. Information is transferred sequentially, bit by bit, from one active station to the next. Each station generally regenerates and repeats each bit. A given station transfers information onto the ring, where the information circulates from one station to the next. The addressed destination station(s) copies the information as it passes. Finally, the station that transmits the information effectively removes the information from the ring.

A station gains its right to transmit its information onto the medium when it detects a token. The token is a control signal comprised of a unique signaling sequence that circulates the ring following each information transfer. At the completion of its information transfer, the station initiates a new token, which provides other stations the opportunity to gain access to the ring.

1.2 Overview of COLAN IV

COLAN IV is one of the four experimental Control Oriented LANs (COLAN) developed at Oregon State University. The first, COLAN, was developed by Yue-Peng Zheng [ZHEN 86]; the second, COLAN II, by Shao-Kong Kao [KAO 87]; and the third, COLAN III, by Doo-Hun Eum [EUM 87].

The physical configuration of COLAN IV consists of several nodes linked into a local area network by a shared communication medium. Each node has two components, a PC and a Network Interface Unit (NIU). Nodes are connected to the shared communication medium through their NIUs (Figure 1.2). Communications within a node, i.e. between the PC and the NIU, utilize the common asynchronous technique of RS-232c. The standard serial port, COM 1, of the IBM PC provides a full-duplex, serial interface with the NIU at the data rate of 9600 baud. Communications among network nodes via their shared communication medium are carried out with a half-duplex, asynchronous, serial, baseband transmission of 9600 baud.

Twisted-pair cable serves as the shared communication medium which interconnects all the nodes with a linear "BUS" network topology. The BUS topology provides a simple but effective way of realizing a decentralized control structure of a LAN. The RS-485 bus standard is adopted to achieve a high performance with a low cost twisted-pair cable. It supports data rates up to 10 Mbps with a 50 ft cable, or 100 Kbps with a 4000 ft cable. At a data rate of 9600 baud, the shared communication medium of COLAN IV can operate with a cable length up to 4000 ft.

The inherent physical parameters of the shared network bus, e.g. the characteristic line impedance, interface impedances, and the sensitivity threshold of receivers, limit the total number of nodes to about 256. However, the current configuration of COLAN IV supports only 63 nodes. Addresses are assigned by the use of an internal DIP switch. The "0" address is reserved as a "universal" address to be used specifically for broadcasting data to all network nodes.

The software for each node was developed separately for the network interface unit (NIU) and the PC. The NIU's software is implemented with the low level assembly code of 8031 while the PC's software is written in Turbo Pascal, a high level language.

Designed to operate under light loads, COLAN IV adopts the CSMA/CD media access method. The CSMA/CD technique outperforms both TOKEN BUS and TOKEN RING techniques in low load environments [HAMM 86]. Furthermore, no complex software algorithm nor complicated hardware is needed to implement the CSMA/CD technique. This simplicity is a favorable feature which makes the network highly reliable and maintainable.

COLAN IV is designed for the dual purpose of communications and control. As such, it is functionally flexible. It may either function as a pure communications network, or as a pure control system, or as a mix of both. As a communications network, the services provided by COLAN IV are

1. electronic mail,
2. resource sharing, and
3. network bus monitoring.

Electronic Mail is the major application of COLAN IV in its capacity as a communications network. Electronic mail provides personal mail boxes in all network nodes and offers the ability to exchange data bases and information electronically among network users.

Resource Sharing is another service provided to all network users. At present, only a simple type of resource sharing has been implemented, i.e. printer sharing. To reduce cost, a single shared printer may be utilized to service the printing needs of the entire local area network.

A minor but important function provided is the Network Bus Monitoring. It is a passive function that captures all the data traffic passing through the network bus. It serves the purpose of network management and maintenance.

Since COLAN IV is a dual purpose LAN, it can also serve as a real-time control system. A complete distributed control system can be set up with a PC serving as the task scheduler and a group of NIUs serving as microcontrollers. In fact, several independent and complete control systems of various sizes may exist simultaneously in the same network. Such a configuration is particularly useful in a campus environment where several researchers may utilize the same network to control and monitor research experiments from their individual offices. The following are some examples of control application achievable with COLAN IV.

1. Control and monitor research experiments, manufacturing processes, sensors, and displays.
2. Control and gather information from specialized educational training devices.
3. Supervise multiple activities and provide a means for gathering, recording, and analyzing the data they produced.
4. Coordinate and synchronize multiple actions at remote sites.
5. Distribute and collect information selectively from multiple remote locations.

Being a dual purpose LAN, COLAN IV can support several control systems while operating as a communications network.

Detailed discussions of COLAN IV are presented in the succeeding chapters. Chapter two presents the design issues of the shared communication medium. Chapter three discusses the hardware implementations. Chapter four describes its functional architecture in terms of four layers. Chapter five explains the working principles of the networking and application software. Finally, chapter six concludes with several suggestions for future improvements of COLAN IV.

Appendix A describes the procedures for setting up COLAN IV and provides operational guide to first time users.

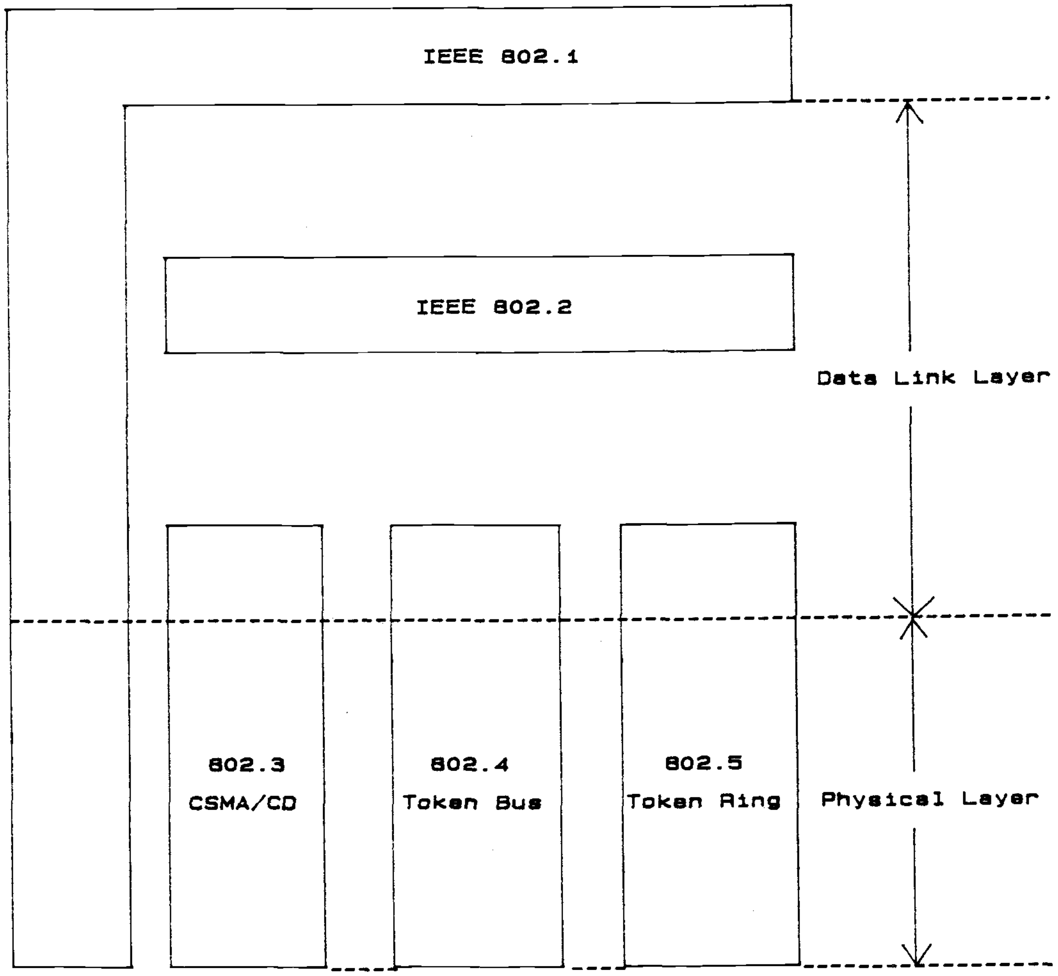

Figure 1.1 IEEE 802 Standards

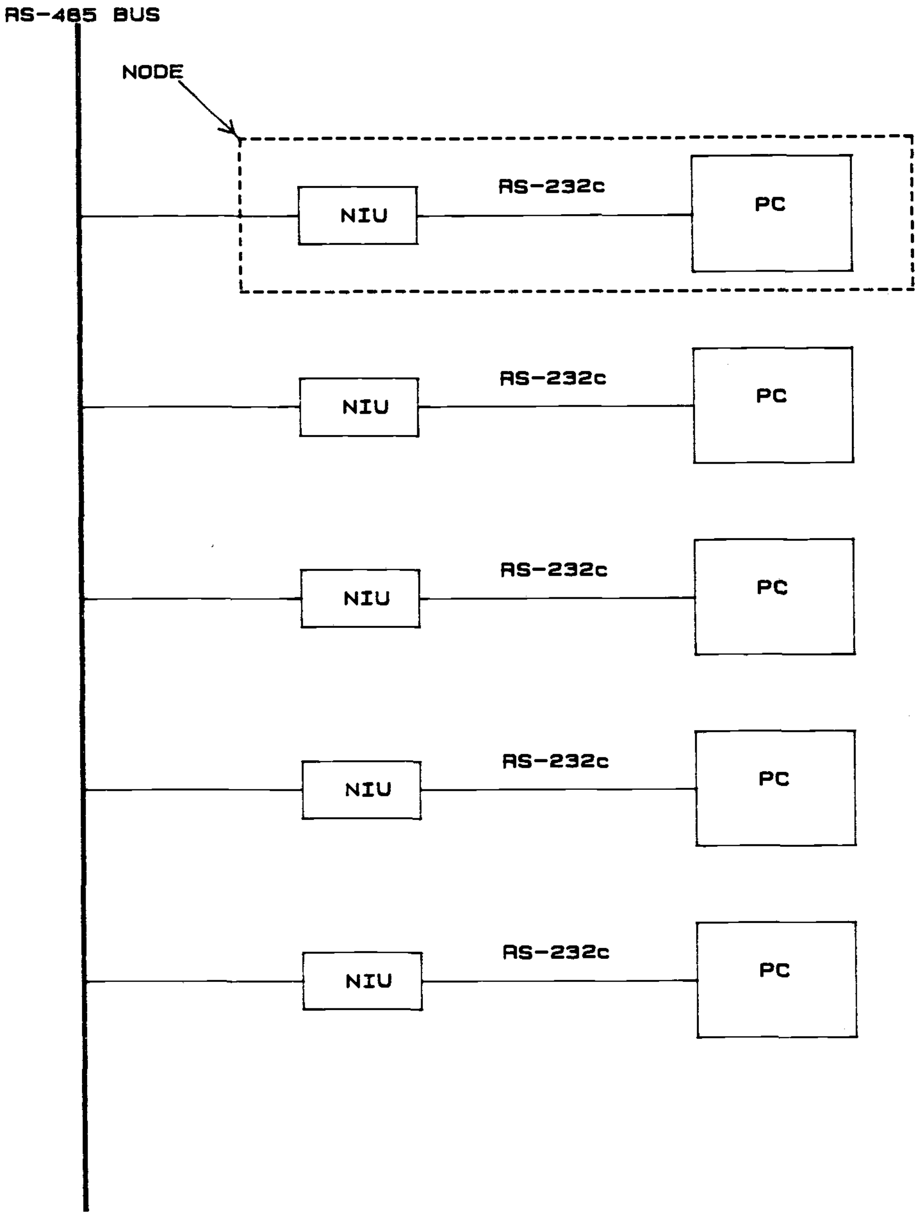

Figure 1.2 COLAN IV Configuration

CHAPTER 2

COMMUNICATION MEDIUM FOR NETWORK INTERCONNECTION

2.1 Overview

The following sections discuss the design issues of the communication medium which links all the nodes into a local area network. Justifications are presented for adopting

1. Bus topology,
2. Twisted-pair cable, and
3. RS-485 Bus standard.

2.2 Bus Topology

The term "topology" refers to the manner in which nodes are interconnected. The topologies currently in common use are mesh, star, tree, ring, and bus. The mesh topology is not attractive for the design of a low cost LAN because it demands complex packet routing software and a high cable cost in providing multiple data routes. The centralized routing control of a master node in the star topology is not suitable for implementing distributed control structure among peer nodes. The tree topology has the drawback that baseband transmission is limited to one direction only, i.e., from root to branches. Although the ring topology is simple in structure, it however requires the active involvement of all the nodes to circulate a message in the network, and hence the failure of any interface in any node will result in a defective network. Bus topology is chosen for these favorable characteristics:

1. It requires no packet routing algorithm.
2. It does not have a centralized master node and hence is suitable for distributed control structure.
3. Nodes can be added or deleted easily.
4. Each node is passive, but has the ability to read the packet header as it passes by.
5. Malfunction in any node will often be localized and will not affect the total network operation.

2.3 Twisted-pair cable

Twisted-pair cable is one of the oldest and most widely used mediums. It has been largely used in telephone communications, and therefore is inexpensive and readily available. The twisting of a pair of wires provides some shielding against interference. Baseband transmissions up to 10 Mbps with 50 ft cable length is possible if a differential transmission technique is used. Although coaxial cable offers better performance it is, however, considerably more expensive and bulky. Twisted-pair cable is chosen for the following reasons:

1. It has a good performance/cost ratio, and hence is cost effective if a differential transmission technique is used.

2. It is readily available.
3. It has a small diameter, and thus can be manipulated easily.

2.4 EIA RS-485 Bus Standard

EIA RS-485 bus standard is adopted because it offers high potential performance while utilizing low cost twisted-pair cable. Reasonably good performance at low cost is made possible by utilizing a differential transmission technique with twisted-pair cable. Interference is initially reduced by the twisting of wires and further suppressed by the differential transmission technique since the common mode interference signals are eventually rejected by the differential line receiver. A desirable feature of RS-485 bus is that both the line driver and receiver operate from a single 5-volt power supply. The RS-485 bus supports a data rate of up to 10 Mbps with a 50 ft cable, or 100 Kbps with a 4000 ft cable. Cable length is limited to 4000 ft for data rate below 100 Kbps. At 9600 baud, COLAN IV can operate with a cable length up to 4000 ft using a 24 AWG twisted-pair telephone cable, with a shunt capacitance of 52.5 pF/meter, terminated in a 100 ohm resistive load.

CHAPTER 3

HARDWARE IMPLEMENTATIONS

3.1 Overview

The hardware of COLAN IV consists of a number of nodes linked into a local area network by twisted-pair cable. The hardware of each node consists of two components, a personal computer (PC) and the hardware of NIU. The following sections describe the hardware implementations in the PC and the NIU.

3.2 Personal Computer (PC)

Information interchange between the two components of a node, i.e., the IBM PC and the NIU, uses a RS-232C serial interface. No additional circuit boards nor modifications are needed in the PC hardware. A standard serial port, COM 1, of the IBM PC is used to provide full duplex, asynchronous serial communication with the NIU at a data rate of 9600 baud.

3.3 Network Interface Unit (NIU)

The hardware of a NIU is completely housed in a single "SIBEC II" circuit board made by Binary Technology Inc.. As depicted by Figure 3.1, each NIU consists of the following hardware components:

1. Microcontroller system

2. NIU/PC serial interface circuit
3. Carrier sense circuit
4. RS-485 interface and collision sense circuit
5. Centronics printer port
6. Node address identification circuit
7. NIU's local serial port

3.3.1 Microcontroller System

The main hardware component of a NIU is the microcontroller system (Figure 3.1). Its CPU is a microcontroller chip of the MCS-51 family, which includes 8031, 8051, and 8751. The microcontroller system runs at the system clock rate of 7.328 MHz. In addition to the CPU chip, the microcontroller system includes:

1. One 8K PROM (2764) for program memory which resides at the addresses from 0000H to 1FFFH.
2. Four 8K RAMs (6264) for data memory which resides at the addresses from 2000H to 7FFFH.
3. One programmable peripheral interface chip (8255) for the Centronics printer port and the node address identification circuit. It resides at the addresses from F000H to F003H.

The on-chip serial port of the microcontroller chip handles the digital communication with other network nodes via the RS-485 interface circuit and the shared network bus. It is programmed to support half-duplex, asynchronous, serial transmission at a data rate of 9600 baud. COLAN IV deploys a special feature of the microcontroller chip to achieve an efficient multiprocessor communications. This feature allows characters to be transmitted with an extra bit added to the normal 8-bit data. The added bit is a special purpose bit which can be programmed to "1" or "0" to differentiate a destination addressing character from an ordinary data character. Destination addresses are transmitted with a special purpose bit of "1" while data characters are transmitted with a special purpose bit of "0."

Information interchanges among network nodes are carried out through data packets whose header includes destination addresses. The NIUs are programmed to allow destination addresses, which have the special purpose bit set to "1", to interrupt the microprocessors. Although all network nodes may be interrupted by the destination address of a packet, only the specifically addressed node continues to receive the data contained in that packet. The other nodes simply ignore the subsequent data for packets not addressed to them. They resume their respective tasks and hence avoid wasting system resources to service interrupts caused by irrelevant data on the network bus. This arrangement improves the overall efficiency of the network.

3.3.2 NIU/PC Serial Interface Circuit

The NIU/PC serial interface circuit (Figure 3.2) includes channel A of an Intel 82530 serial communication controller (SCC), a line driver (1488), a line receiver (1489), and a DB-25 connector. Efficient interchange of information between the SCC and the microcontroller chip is handled by the external interrupt (EX1) service routine. This circuit provides a full duplex, asynchronous serial communications between the NIU and the PC at a data rate of 9600 baud.

3.3.3 Carrier Sense Circuit

The carrier sense circuit continually monitors the RS-485 network bus. The CSMA/CD media access method relies on this circuit for the management of media access control. It consists of a retriggerable monostable chip (74LS123), which is triggered by the one-to-zero transition of digital signals (Figure 3.3). Each triggered pulse has a duration of 15 ms. This duration is sufficient to accommodate either the collision recovery process or the acknowledgement process after receiving an error free data packet. Since the monostable chip is triggered at least once by each character, its output indicates the presence or absence of digital signal on the network bus.

3.3.4 RS-485 Interface and Collision Sense Circuit

The circuit that provides the RS-485 interface and the Collision Sensing consists of a differential line receiver (75175), a differential line driver (75174), two diodes, and two resistors (Figure 3.3).

The tri-state control of the line driver (75174) is driven by the CPU of the microcontroller system. The line driver is turned-on only when its node is transmitting data into the RS-485 bus. Otherwise it is tri-stated (high impedance) to prevent unnecessary loading of the RS-485 bus by its low output impedance. The line driver and receiver constitute the RS-485 interface circuit. The addition of two diodes and two resistors enables it to detect a collision while transmitting data. The circuit configuration creates a "1" dominance bus, i.e., the signal on the bus is "1" if any node transmits a "1" even though other nodes may simultaneously transmit a "0." To detect a collision the node's line receiver reads the data on the bus while its line driver is transmitting. A collision is detected if the received signal does not match the transmitted signal.

3.3.5 Centronics Printer Port

The Centronics printer port consists of the port B and port C of a programmable peripheral interface chip (PPI) and a DB-25 connector (Figure 3.4). The PPI is programmed to provide a functional configuration in which port A and port B operate as dual 8-bit I/O ports while port C handles "strobe" and "handshaking" signals.

Port B is programmed to provide an 8-bit parallel interface to a standard printer. Port C not only provides the necessary "strobe" signal that accompanies an output character to the printer, but also detects the "busy" and the "acknowledge" signals from the printer. The operating system in the NIU uses the "busy" signal to control the output of data to the printer. It refrains from sending a character to the printer whenever the "busy" signal from the printer goes high. But when the "busy" signal goes low, it initiates the output of an 8-bit character to the printer via the port B as well as an accompanying "strobe" signal via port C.

The "acknowledge" signal has not been used because a simple but effective printing algorithm can be achieved with the use of the "busy" signal alone.

3.3.6 Node Address Identification Circuit

The network address of each node is specified by the 6 bits of a switch bank commonly known as a DIP switch (Figure 3.5). The 6-bit DIP switch defines an address range of 0 to 63, or 0H to 3FH in Hexadecimal. The network address, as specified by the DIP switch settings, is supplied to the microcontroller chip via port A of programmable peripheral interface (8255). Since port A has 8 I/O lines the addressing range can be expanded to support 256 unique addresses if an 8-bit DIP switch is used.

3.3.7 NIU's Local Serial Port

A local serial port is installed in the NIU in addition to the NIU/PC serial interface (Figure 3.2). It consists of a DB-25 connector and the channel B of the serial communication controller (82530). Its presence allows the utilization of a wide assortment of serial devices. The software in the NIU provides a simple protocol ("XON" and "XOFF") for data flow control. It may be used to provide a gateway to a modem so that the network can be accessed from a remote location through telephone lines.

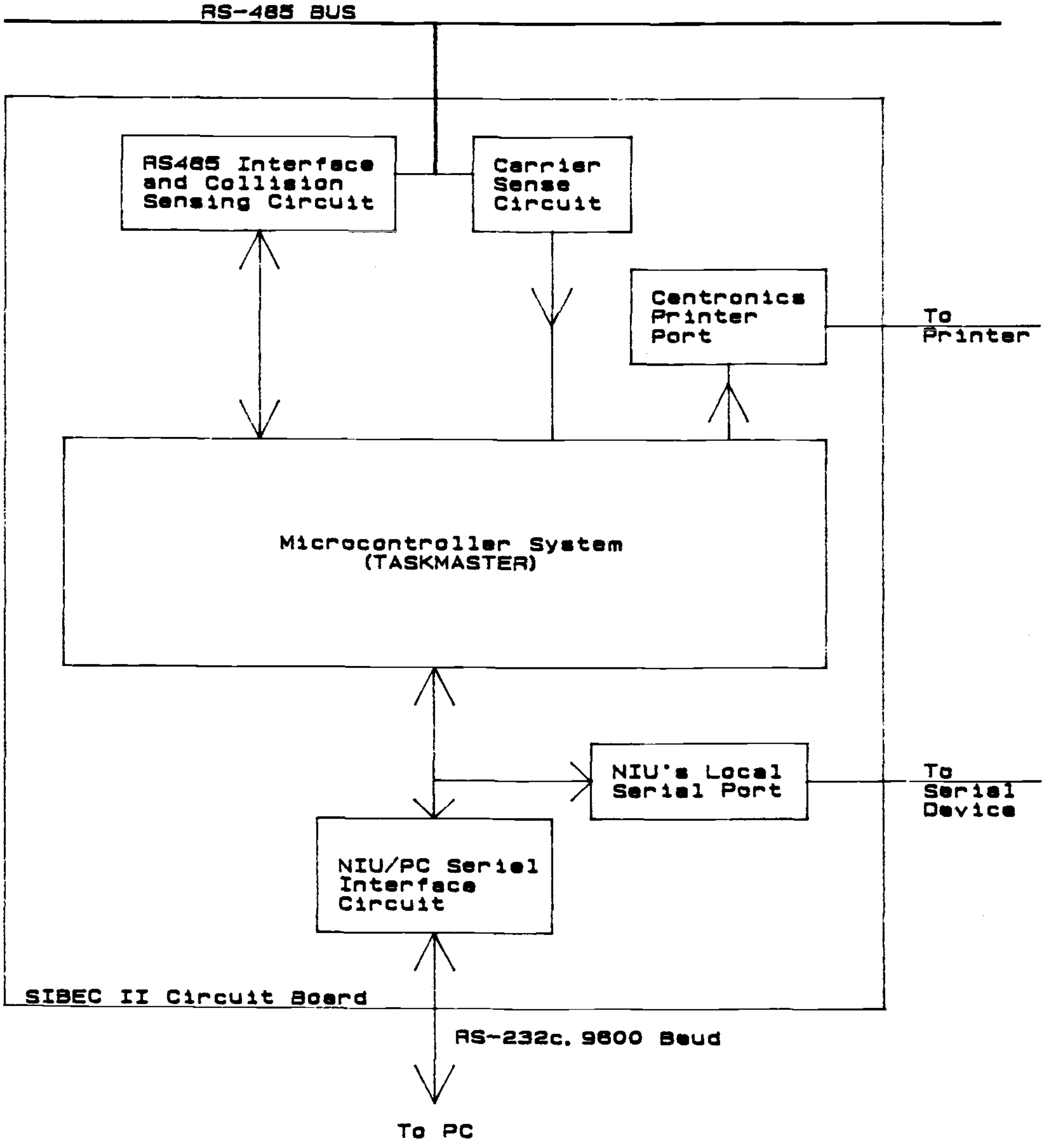

Figure 3.1 Hardware Components of a Network Interface Unit

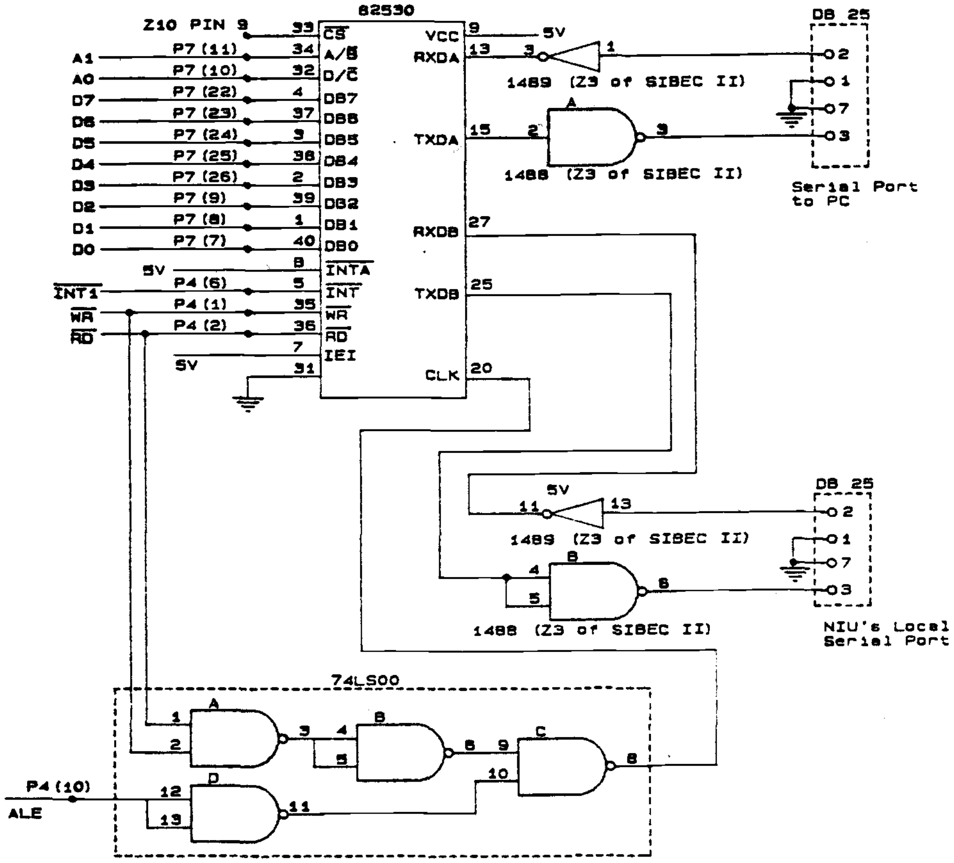

Figure 3.2 NIU/PC Serial Interface and NIU's Local Serial Port.

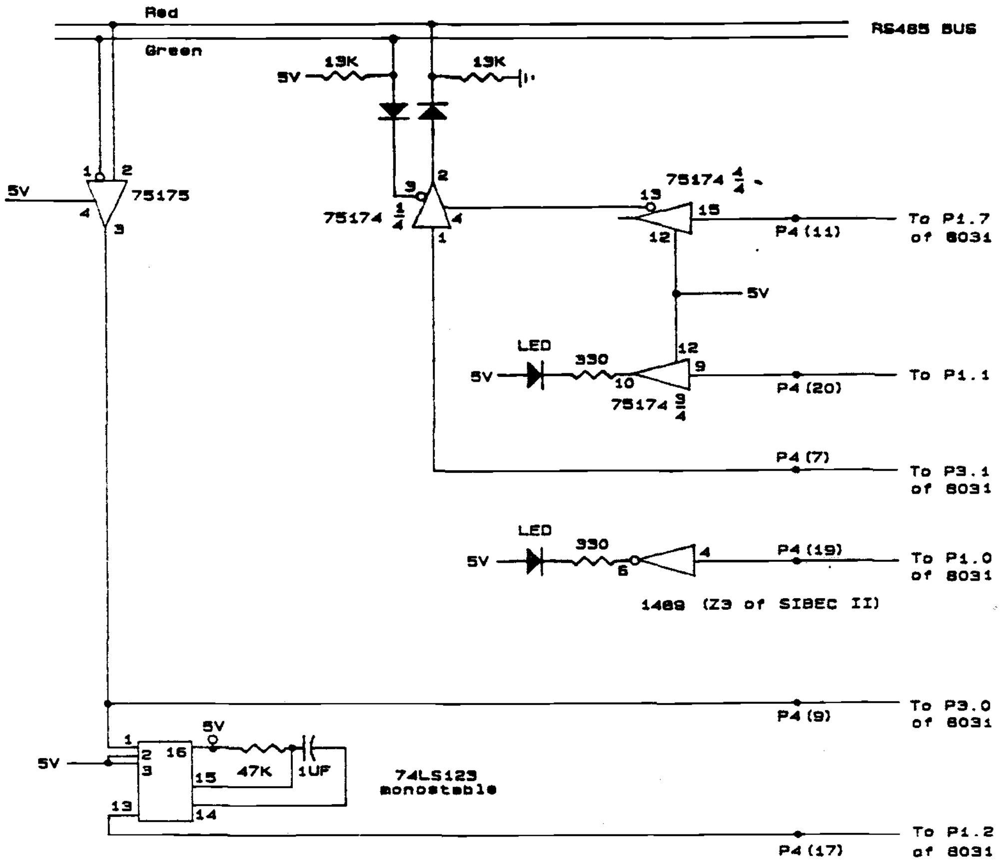

Figure 3.3 Carrier Sense Circuit, RS-485 Interface Circuit, and Collision Sense Circuit.

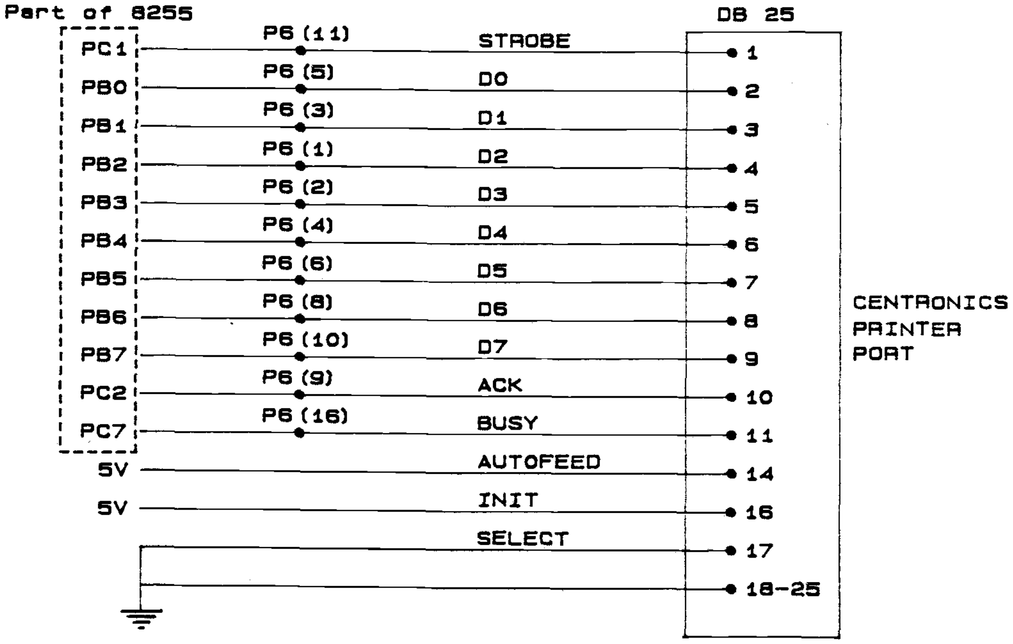

Figure 3.4 Centronics Printer Port

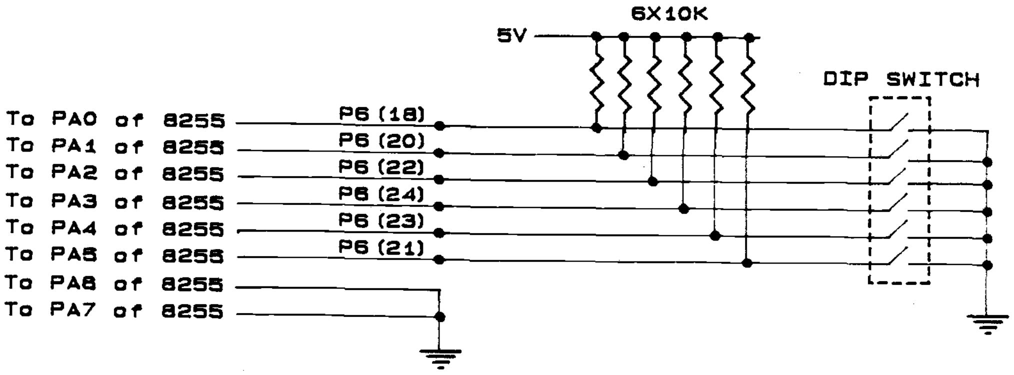

Figure 3.5 Node Address Identification Circuit

CHAPTER 4

FUNCTIONAL ARCHITECTURE OF COLAN IV

4.1 Overview

The ISO open systems interconnect reference model has been adopted in the design of the functional architecture of COLAN IV. The OSI reference model, which specifies seven functional layers, is not fully implemented. The network layer has not been implemented because it is not necessary for a LAN in which a single bus links all nodes. The session layer is not required since the system is not designed to handle multi-session operations. The presentation layer is not implemented because code conversion and data reformatting are not necessary in COLAN IV. The four functional layers implemented (Figure 4.1) are

1. Application layer,
2. Transport layer,
3. Data link layer, and
4. Physical layer.

User applications and services provided through the application layer includes: distributed control systems, electronic mail, resource sharing, and network bus monitoring. Network users can access these services via a "friendly" human interface provided by the software which resides in the PC. Data generated in the application layer such as the destination address, the task command, the user's data, etc., are presented to the transport layer. These data are initially processed in transport layer and then further reformatted in the data link layer before they are transmitted as a data packet by the physical layer. In the receiving end, the data packet is sequentially processed as it threads through the data link layer and the transport layer before its intended application is finally serviced in the application layer.

A simple form of transport layer has been implemented to allow the transparent transfer of data between network users, and the buffering of data to regulate the flow of information. Data flow through the transport layer in two directions, outgoing and incoming. They are each treated in a different manner. Outgoing data supplied by the application layer are processed and assembled into a data packet before they are forwarded to the data link layer. Incoming data packets, received from the data link layer, are disassembled, processed, and delivered to the application layer.

Data packets also flow through the data link layer in both directions. In the data link layer, outgoing and incoming data packets are respectively processed and reassembled into appropriate forms before they proceed to the next layer. In addition to the processing of data packets, the data link layer also manages CSMA/CD media access control and logical link control. The functions of the data link layer are carried out by two sublayers, i.e. the logical link control sublayer (LLC) and the media access control sublayer (MAC). The LLC manages two possible types of data links: the connection-oriented data link and the unacknowledged connectionless data link. The MAC manages CSMA/CD media access control.

Both the transmission and reception of data by the physical layer via the network bus are performed under the control of the MAC sublayer.

The following sections present detailed discussions of individual layers.

4.2 Application Layer

The software of the application layer resides partly in the PC and partly in the network interface unit (NIU). The main services provided by the application layer are

1. Distributed control system,
2. Electronic mail,
3. Resource sharing, and
4. Network bus monitoring.

4.2.1 Distributed Control System

Complex control and monitoring systems often require multiple microcontrollers, each optimized and located for the best performance [HERZ 87]. In order to achieve overall control and synchronization, a single system scheduler may be used to manage a group of microcontrollers. This control structure can be compared to a simple two level production organization where the system scheduler functions as the "manager" who makes higher level decisions concerning the production quantities. Commands are transmitted to the skilled and specialized production work force (the microcontrollers) for coordinated action. Any NIU in the network can be utilized as a controller since each has an enhanced TASKMASTER.

The TASKMASTER is an experimental microcontroller system which has been designed at Oregon State University to perform real-time, distributed control operations. Its operating system constitutes the core of the software in the NIU. TASKMASTER systems were originally designed to operate with a master-slave control structure utilizing a daisy-chain connection. A distributed control system set up by TASKMASTER systems requires a PC to serve as the master which manages communications within the control system. COLAN IV abandons the undesirable master-slave control structure but adopts the task oriented operating system of the TASKMASTER. The capabilities of the TASKMASTER's operating system have been enhanced and modified to provide major functions of the software in the NIUs of COLAN IV.

Within the COLAN IV environment, a complete distributed control system can be set up by using a PC as the system scheduler and a group of NIUs as controllers (Figure 4.2). In fact, several independent and complete control systems of various sizes may exist simultaneously in the same network.

Such a configuration is particularly useful in an environment where several researchers may utilize the same LAN to control and monitor research experiments from their individual offices. Each researcher may set up an independent distributed control system using a PC in the office as the system scheduler and a group of NIUs in the laboratory as microcontrollers.

Task Oriented Control Structure

The operation of the control system is divided into "tasks." A task is an independently scheduled and programmed event which requires and services system resources [HERZ 87]. In the case of a controller, the resources are the various pieces of hardware used in the specialized control and monitoring activity. These resources are serviced through the use of subroutines specifically written for the microcontroller system and its application environment. A task, then, is a modular piece of software used to accomplish a specific action. By the proper sequencing of tasks, complex control and monitoring activities can be carried out.

Four pieces of information are necessary to specify a task of the microcontroller system.

1. **Name:** The name that identifies the appropriate software subroutine.
2. **Starting Mechanism:** The method, or conditions under which the task is to be started. In a simple system, a task

may commence when it reaches the head of the task queue. Some tasks may require "synchronization" from the system scheduler before they can begin execution.

3. **Terminating Mechanism:** The method, or conditions under which the task will be terminated or aborted. In a simple system a task may run until completion. In some instances, tasks may be aborted under command from the task scheduler or time out. A task may be discarded after termination, or it may be requeued for repeated running.
4. **Parameters:** A task may require specialized data or further specification before it can run.

User Interface for Distributed Control System

The software of the system scheduler which provides the user interface resides in the PC of each node. It is designed to give a friendly menu approach for assisting and guiding the user through all the necessary operating steps. The scheduling of tasks for a control system can be easily accomplished through two menu items:

1. Local Tasks
2. Remote Tasks

Local Tasks

When the item "Local Tasks" is selected from the menu, the PC displays a listing of all the local tasks available to the user. As its name implies, the user may only schedule tasks which involve the resources within the local node.

Local task command packets (LTCP) are protocol data units used to accomplish the execution of local tasks. The system scheduler generates the appropriate LTCPs and transmits them via the RS-232c link to the local NIU for execution. The following diagram illustrates the format of a LTCP.

Format of LTCP	Example
start delimiter	{
device number	0 0
prefix	!
task number	0 B
postfix	.
task parameters ⋮	0 2 A B
end delimiter	}

Start delimiter: The left brace "{" is used as a start delimiter.

Device number : A sequence of 2 ASCII Hexadecimal numerals. Since a local command is confined within a node, the universal address 00 provides a simple means of addressing the local NIU.

Prefix : An ASCII character which specifies the starting mechanism.

Task number : Two ASCII Hexadecimal characters to specify the software subroutine.

Postfix : An ASCII character which specifies the terminating mechanism.

Parameters : Zero to 5 pairs of ASCII Hexadecimal characters required for the execution of the task.

End delimiter : The right brace "}" is used as an end delimiter

Remote Tasks

When the menu item "Remote tasks" is selected, the PC responds with a listing of all the remote tasks available to the user. This service enables the users to schedule tasks in any remote node that is connected to the LAN. Complete control and monitoring functions of a distributed control system can be easily accomplished through the combination of remote tasks and local tasks.

Network task command packets (NTCP) are protocol data units used by remote tasks. Beside generating the appropriate NTCP, the system scheduler also receives a destination address from the user through the keyboard input. It delivers both the NTCP and the destination address to the transport layer which assembles them into a data packet. The resulting data packet is further processed in the data link layer before it is transmitted to the destination node by the physical layer. In the destination node, it threads through the three lower layers and eventually initiates an appropriate action in the application layer.

The NTCP has the same data format of a LTCP, except the device number field is removed. The following diagram illustrates the format of a NTCP.

Format of a NTCP

start delimiter
prefix
task number
postfix
task parameters : :
end delimiter

Example

{
!
2 6
.
}

4.2.2 Electronic Mail

Another service provided by COLAN IV is the electronic mail. Electronic mail offers the ability to exchange data files and information electronically among network users. Electronic mail allows the sending and receiving of messages through electronic mail boxes.

Conventional mail boxes are utilized to transmit messages in situations where the intended recipient cannot be contacted by telephone. But they have some disadvantages. For example, the mail boxes may not be conveniently located. The same inconvenience applies to the recipient who has to pick up the message. Electronic mail, on the other hand, provides personal mail boxes which are conveniently located in the users' offices.

User Interface for the Electronic Mail

The software of the electronic mail of each node resides partly in the PC and partly in the NIU. The personalized electronic mail box of each user is physically located in the NIU. A dip-switch within each NIU is configured to establish a unique network address.

The electronic mail feature of COLAN IV employs a friendly, easy-to-use menu approach for assisting the user in the information exchange operations. Its menu displays possible alternatives and prompts the user through all necessary steps. Electronic mail services can be accessed via its four menu items.

1. Mail directory
2. Read Mail
3. Purge Mail
4. Transmit mail

Mail Directory

When the item "Mail directory" is chosen from the menu, a listing will be displayed on the screen, indicating the number of messages that have been received, the senders of individual messages, and the memory locations where they are stored.

Read Mail

The network user can randomly access individual messages that have been received into the personal mail box, and may elect to browse a selected message on the screen or have it written to the disk memory.

Purge Mail

This function of electronic mail allows the user to purge the electronic messages that have been processed and are no longer needed in the personal mail box.

Transmit Mail

A message may be personal or public. A personal message is to be transmitted to a single intended recipient by specifying the unique network address. A public message, which is intended for all network users, will be broadcast using the universal address. Currently the network supports a total of 64 addresses, i.e., 00 to 63. Address 00 is assigned as the universal address.

In order to send a message, a network user begins by specifying the network address of the recipient. A menu will be displayed in response, offering four options.

1. Send message to mail box of recipient.
2. Send message to mail box and printer of recipient.
3. Send message to mail box and local serial port of recipient.
4. Send message to mail box and PC of recipient.

After choosing one of the options above, the user will be further prompted to select one of the two alternatives:

1. Send a short message which can be created by typing it on the PC screen (up to 700 Bytes).
2. Send a message which has previously been created and stored as a disk file (up to 8 KByte).

The outcome, i.e. the success or failure of transmitting the message to the recipient, will be reported to the user. Any ASCII file may be transmitted as a message.

4.2.3 Resource Sharing

At present, only simple resource sharing has been implemented, i.e. printer sharing. The entire network may share a single printer. Since every NIU has been equipped with a Centronics printer port, the single shared printer can be connected to any NIU which is conveniently located. Several network nodes may send their files to the NIU which is equipped with a shared printer and have their hardcopies printed sequentially on a first-come-first-served basis. Printing activities of the shared printer are carried out in the background transparent to network users.

The background printing is said to be transparent because it neither affects the performance of the LAN nor the normal operation of the particular node which does the background printing with the shared printer. Background printing activity will be further discussed in chapter 5.

4.2.4 Network Bus Monitoring

The network bus monitoring function is available to every node for the purpose of network management and maintenance. It is a passive dedicated function that captures all the data traffic passing through the network bus and, if necessary, records them in disk files. Troubleshooting of network problems can be done by analyzing the captured data-traffic that flows through the bus.

4.3 Transport Layer

A simple form of transport layer has been implemented. It provides the transparent transfer of data between network users, and the buffering of data to regulate the flow of information. Data flow through the transport layer in two directions, outgoing and incoming. When the application layer of one node communicates with its peer application layer of another node, the transport layer in the transmitting node operates in "output mode," while the identical, peer transport layer in the receiving node operates in the "input mode." Data are processed in different manners in these two operational modes.

4.3.1 Output Mode of Transport Layer

The transport layer of a transmitting node operates in "output mode." The information it receives from the application layer includes the destination address, the source address, the network task command packet (NTCP), and the user's data. A transport packet (TP) is assembled out of these data. The following diagram illustrates the format of a TP.

TP Format	Example
start delimiter	[
destination address	1 8
source address	0 2

NTCP	{
	:
	2
	6
	.
	}

: user's data :	1 : 3
end of file	EOF
end delimiter	]

The individual elements of the transport packet are:

Start delimiter : "[" is used as the start delimiter.

Destination address: Two ASCII characters representing a Hexadecimal number ranging from 00H to 3FH.

Source address : Two ASCII characters representing a Hexadecimal number ranging from 00H to 3FH.

NTCP : Network task command packet.

User's data : 0 to 8 KByte of ASCII data are allowed.

End of file : ASCII character, 1AH in Hexadecimal.

End delimiter : "]" is used as the end delimiter.

The transport layer, in the output mode, is responsible for placing the transport packet in the appropriate buffer where it can be accessed by the data link layer for further processing.

4.3.2 Input Mode of Transport Layer

In input mode, the transport layer receives data from data link layer in the form of Logical Link Packets (LLP). These packets will be disassembled into source address, NTCP, and user's data before they are forwarded to the application layer. The following diagram illustrates the format of a LLP.

LLP Format	Example
start delimiter	[
source address	0 2
NTCP	[: 2 6 . 0 1]
: user's data :	1 : 3
end of file	EOF
end delimiter	]

LLP has the data format of a transport packet in which the destination address field is omitted. Source address, NTCP, and user's data are the three components extracted from the LLP by the transport layer. They are forwarded to the application layer to effect appropriate action.

4.4 Data Link Layer

The functions of data link layer are carried out by two sublayers, the Logical Link Control sublayer (LLC) and the Media Access Control sublayer (MAC). The software of the data link layer resides entirely in NIU. A "data link" is an assembly of two or more network nodes and the interconnecting communications channel in which information is exchanged. The data link layer is the conceptual layer responsible for maintaining the control of the data link. Its functions provide an interface between the transport layer and the physical layer. These functions include address field interpretation, channel access, generation of logical link packet (LLP) and media access packet (MAP).

Two types of data link control operations are supported. The first type provides an unacknowledged connectionless link for broadcasting data to all network nodes using the universal address of 00. No acknowledgement is expected from the receiving nodes. The second type provides a connection-oriented link for transmitting data to a single destination. The receiving node is expected to return an acknowledgement if no data error is detected, otherwise, it remains silent.

4.4.1 Logical Link Control(LLC) Sublayer

The logical link control (LLC) sublayer provides a functional interface between the media access control (MAC) sublayer and the transport layer. The functions performed by the LLC sublayer can be described in terms of two types of data link control operations.

1. Connection-oriented link
2. Unacknowledged connectionless link

Connection-Oriented Link

Transmitting node

The LLC sublayer of the transmitting node accepts data from the transport layer in the form of a transport packet (TP), which is buffered in the NIU. It forwards the transport packet to the MAC sublayer for transmission to a single destination. It expects an acknowledgement from the recipient at the end of transmission. If an acknowledgement is not returned within a specified time, it then schedules the retransmission of the same packet. The mission is considered a failure if it receives no acknowledgement after three attempts. The outcome, success or failure, will be reported to the user.

Receiving Node

In the receiving node, the LLC sublayer will generate an acknowledgement (06H in Hexadecimal) if the MAC sublayer reports that a complete packet has been received without error. If an error is detected, it simply discards the data packet and remains silent. Error free data will be assembled into a logical link packet (LLP) in the LLC sublayer. A completely assembled LLP will be handed over to the transport layer for further processing.

Unacknowledged Connectionless Link

Transmitting Node

Unlike the case of the connection-oriented link, the LLC sublayer of the transmitting node makes only one attempt to broadcast a packet using the universal address of 00. No acknowledgement is expected.

Receiving Node

Since it is an unacknowledged connectionless link operation, the LLC of all receiving nodes will not acknowledge the reception of a complete packet even if no error has been detected. Data packets with error are simply discarded. An error free data packet will be assembled into LLP and delivered to the transport layer.

4.4.2 Media Access Control (MAC) Sublayer

The media access control (MAC) sublayer supports the control functions of the CSMA/CD media access method. The media access procedures include encapsulation/decapsulation of data packets, error checking, and acquisition of the data transmission medium. The services provided by the MAC sublayer allow the local LLC sublayer entity to exchange data units with its peer LLC sublayer entity in another node. The main functions performed by the MAC sublayer are:

1. Data encapsulation/decapsulation of MAP.
   (a) Addressing (handling of source and destination addresses)
   (b) Error detection (detection of transmission errors)
2. Media access management
   (a) Medium allocation (collision avoidance)
   (b) Contention resolution (collision handling)

Format of a Media Access Packet (MAP)

The MAP is encapsulated/decapsulated in the MAC sublayer. The MAC converts the destination address of the TP from ASCII format to Hexadecimal format. For example, the destination address of 0BH is represented by two bytes, 30H and 42H, in ASCII format. It will be reduced to one byte, 0BH, in Hexadecimal format. The destination address, now in Hexadecimal format, is duplicated and placed into the first two bytes of the MAP. The duplication of destination address provides a means to prevent any addressing error due to a collision which damages the addressing field of a MAP. The two destination addresses are no longer identical if the address field is damaged by a collision. A damaged address field will be ignored by all network nodes.

The data format of a MAP consists of eight elements, i.e., destination address, start delimiter, source address, NTCP, user's data, ASCII end-of-file character, checksum, and the end delimiter. An illustration of MAP format is shown in the following diagram.

Format of MAP	Example
dest. address	0BH
dest. address	0BH
start delimiter	[
source address	0 2

NTCP	{
	:
	2
	6
	.
	}

: user's data :	1 : 3
end of file	EOF
checksum	A3H
end delimiter	]

The individual elements of the MAP are:

Destination address: One byte Hexadecimal number (0 to 3F)

Start delimiter : "[" is used as the start delimiter.

Source address : Two ASCII characters representing a Hexadecimal number ranging from 00H to 3FH.

NTCP : Network task command packet.

User's data : 0 to 8 KByte of ASCII data are allowed.

End-of-file : An ASCII character (1AH in Hex).

Checksum : A Non-ASCII character obtained by byte-wise Exclusive-Or operation.

End delimiter : "]" is used as the end delimiter.

Operation of the MAC sublayer

The operation of the MAC sublayer will be described for three circumstances: transmission without contention, reception with contention, and access interference and recovery.

Transmission without contention

MAP Encapsulation

When the logical link control (LLC) sublayer requests the transmission of a packet, the MAC sublayer assembles a media access packet (MAP) from the data supplied by the transport packet (TP). It converts the destination address of the TP from ASCII format to Hexadecimal format. For example, the destination address of 06H is represented by two bytes, 30H and 36H, in ASCII format. It will be reduced to one byte, 06H, in Hexadecimal format. The destination address, in Hexadecimal format, is duplicated and placed into the first two bytes of the MAP.

The start delimiter "[" which occupies the first byte of transport packet is repositioned to the third byte of the MAP. A checksum for error detection is appended to MAP. The checksum is evaluated by performing a byte-wise Exclusive-Or operation on all the data, except the destination address, of the MAP. The fully assembled MAP is then forwarded to the Transmit Media Access Management for transmission.

Transmit Media Access Management

The MAC sublayer avoids contention with other traffic on the communication medium by monitoring the carrier sense signal provided by the physical layer. When the medium is clear, packet transmission is initiated. The physical layer, under the control of the MAC, transmits data in the form of baseband digital signals.

Reception without contention

Receive Media Access Management

Data of the MAP on the communication medium are collected by the physical layer under the management of the MAC sublayer. They are immediately handed over to the Receive Data Decapsulation function of the MAC sublayer.

Receive Data Decapsulation Function

The first two bytes of the MAP are destination addresses. They are examined to decide whether the complete MAP should be received. If the destination address does not match its node address, it promptly instructs the Receive Media Access Management to stop collecting data of the current MAP. If the destination address matches its node address then the reception of data is allowed to continue until completion.

It also performs error detection and reports the result to the logical link control (LLC) sublayer. The complete MAP, regardless of error, is delivered to the LLC sublayer. It is the responsibility of the LLC to decide whether an acknowledgement should be returned.

Access Interference and Recovery

If several nodes attempt to transmit at the same time, it is possible for them to interfere with each other, in spite of their attempts to avoid this by deferring. When transmissions from two nodes overlap, the resulting contention is called a collision. If a collision is detected, the Transmit Media Access Management terminates the current transmission and schedules another transmission attempt after a backoff period. To avoid further collision, the backoff period is made proportional to its node address. The entire media access procedures are repeated until it successfully gains control of the communication medium.

4.5 Physical Layer

This section describes the mechanical configuration, electrical signal characteristics, and the procedures to establish, maintain, and terminate physical connection between two or more network nodes. Functions of the physical layer are largely carried out by the hardware with some help from its supporting software which resides in the NIU.

If a node is transmitting information, its physical layer is responsible for the injection of digital signals into the communication medium. On the other hand, the physical layer of a receiving node converts the electrical signals into the corresponding digital data.

The following sections discuss the mechanical configuration, the electrical signal characteristics, functions, and the operating modes of the physical layer.

4.5.1 Mechanical Configuration

The hardware of the physical layer is mechanically coupled to the communications medium by a common, inexpensive telephone jack. Standard 4-conductor telephone wire is used for data transmission.

4.5.2 Electrical Signal Characteristics

Data are transmitted serially and asynchronously via the communication medium in baseband digital signals with signal levels of 0 and 5 volts at a data rate of 9600 baud. The differential transmission technique of the RS-485 bus standard is adopted to achieve a high performance with low cost twisted-pair cables.

The bit stream of digital signals has a non-return to zero (NRZ) signal structure. They are organized into blocks of bits called characters. Each character consists of 11 bits, i.e. a start bit, 8 data bit, a special purpose bit, and a stop bit. The special purpose bit can be programmed to either "1" or "0" to differentiate two types of characters: the data character and the destination addressing character. The special purpose bit is a convenient feature provided by the microcontroller chip of the NIU to support multiprocessor communications.

Media access packets (MAP) are data packets transmitted on the communications medium. Each MAP contains two type of characters. Its first two characters are destination addressing characters with the special purpose bit set to 1. The rest of the MAP are data characters with the special purpose bit cleared to 0.

4.5.3 Carrier Sense Function

A retriggerable monostable logic circuit is responsible for the continuous carrier sensing function. The carrier status, i.e., the presence or absence of digital signals on the communications medium, is presented to the MAC sublayer for the purpose of media access management.

4.5.4 Collision Sense Function

A combination of hardware and software performs the collision sense function whenever a node transmits data into the shared communications medium. In order to detect a collision while transmitting data, it simultaneously reads back the data from the communication medium. A collision is detected if the data it reads back differs from the transmitted data. The collision status, i.e., the presence or absence of a collision, is presented to the MAC sublayer for media access control.

4.5.5 Output Mode of Physical Layer

Although the actual transmission occurs in the physical layer, it is managed by the MAC sublayer of the data link layer.

Information is transmitted, character by character, to the recipient. There are two types of characters: the destination address with its special purpose bit set to 1, and the data character with special purpose bit cleared to 0. Data packets are headed by two identical destination addresses followed by ordinary data characters.

After system reset, an internal flag, SM2, of the microcontroller chip in the NIU is set to 1. In this configuration, i.e. SM2=1, the microcontroller chip responds to destination address only. All passing data characters are ignored.

When a node attempts to communicate with another node, its physical layer switches into "output mode" if the network bus is free. Under the management of the MAC, it transmits a data packet with two leading destination addressing characters which are identical. All network nodes are interrupted by these two addressing characters. Each independently determines whether the packet is addressed to it. The addressed node, upon receiving two valid and identical destination addresses, switches into "input mode" to receive the rest of the packet. This is done by clearing the SM2 flag to 0. The other nodes, which are not addressed, ignore all subsequent data so that their respective systems can resume their interrupted tasks without further interruptions. However, they will be interrupted by the next destination address which accompanies a new data packet. Their SM2 flags remain set to 1.

A silent interval of 15 msec that follows the end of a packet transmission is reserved for the return of an acknowledgement character (06H) from the receiving node. If no acknowledgement has been returned within the silent period, the LLC sublayer schedules a retransmission. The destination node is considered not ready if it fails to acknowledge three consecutive transmissions.

4.5.6 Idle Mode of Physical Layer

The physical layer operates in "idle mode" immediately after system reset. In "idle mode" the physical layer only responds to the arrival of a destination addressing character since the SM2 flag is set to 1.

4.5.7 Ready Mode of Physical Layer

A node which wishes to transmit a packet will be switched into "ready mode" by its MAC sublayer if the network bus is busy. The MAC sublayer continually monitors the communication medium and immediately switches the physical layer into "output mode" when the medium is free.

4.5.8 Input Mode of Physical Layer

The physical layer operates in "input mode" after it has received two valid and identical destination addresses that either matches its node's address or the universal address, 00. To prevent an addressing error due to damaged destination addresses, the MAC sublayer must verify that both destination addresses are identical.

In "input mode" the physical layer is instructed by the MAC to receive all types of characters, i.e., both data and destination addresses. This is done by clearing the SM2 flag to 0. The MAC sublayer switches the physical layer to "idle mode" after a complete packet has been received, and set SM2 flag to 1. The LLC sublayer of the data link layer decides whether an acknowledgement should be returned.

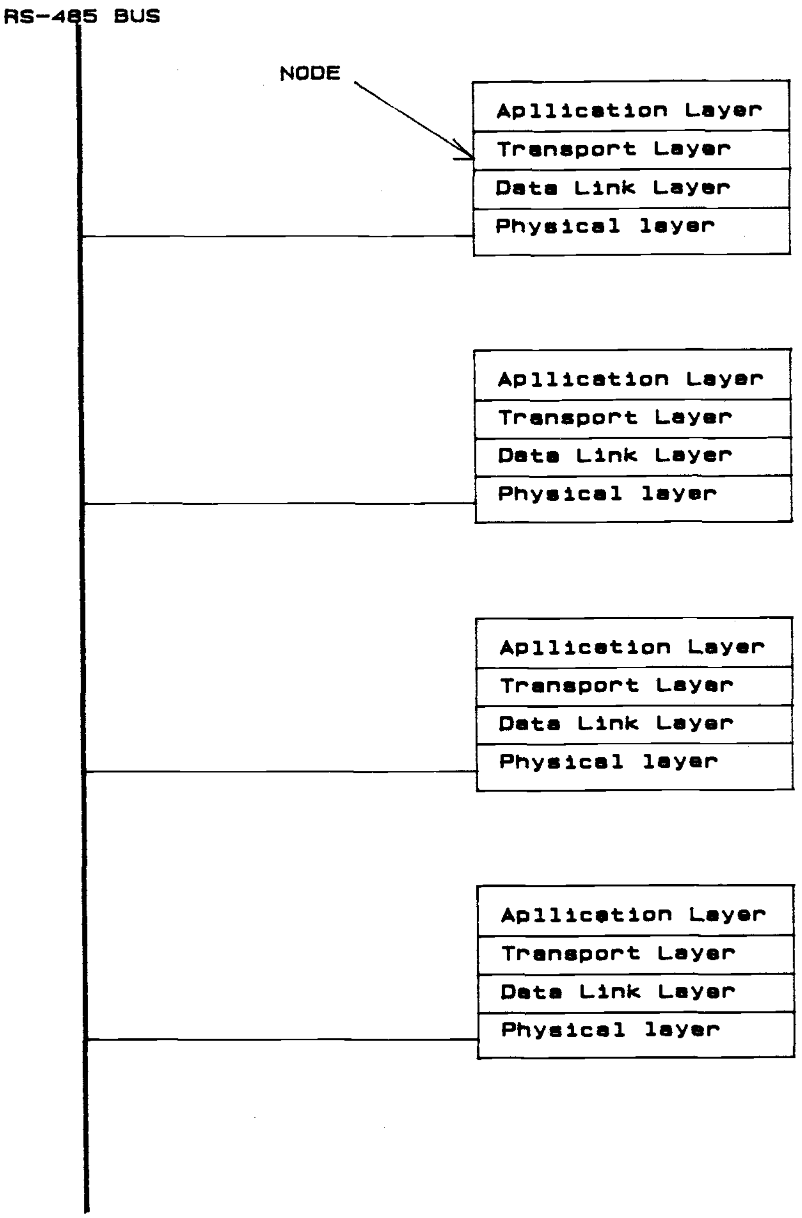

Figure 4.1 Functional Architecture of COLAN IV

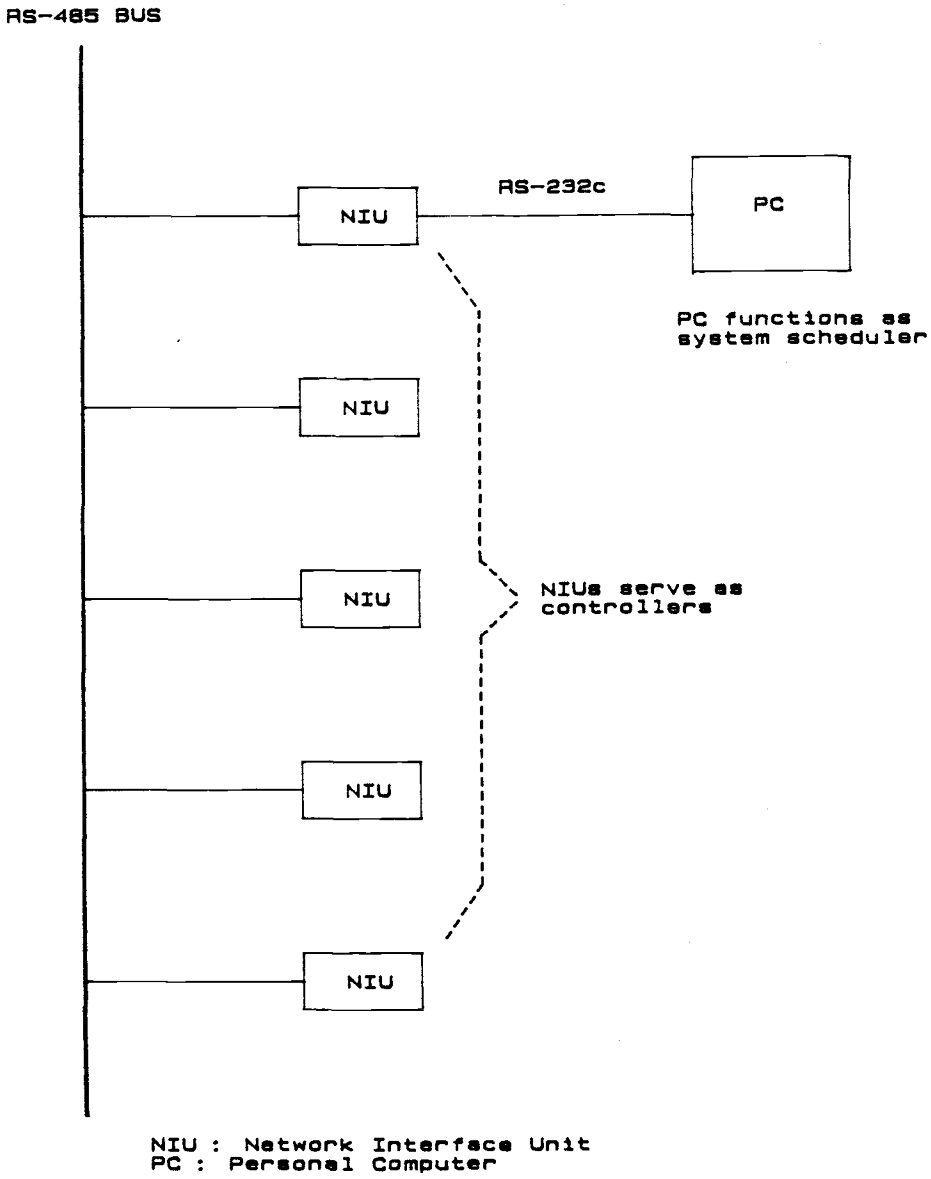

Figure 4.2 A Distributed Control System

CHAPTER 5

SOFTWARE IMPLEMENTATIONS

5.1 Overview

The software of COLAN IV consists of two parts. One part resides in the PC while the other resides in the NIU. These two parts communicate through the RS-232C serial link. The software in the PC is written in Turbo Pascal, a high level language, while the software in the NIU is written in the low level assembly code of the 8031 microcontroller.

The software implementations of COLAN IV will be discussed in terms of the functions performed by the software in the PC and the software in the NIU.

Software in PC

(a) performs some functions of the application layer

(b) performs some functions of the transport layer

Software in NIU

(a) performs some functions of the application layer

(b) performs some functions of the transport layer

(c) performs all functions of the data link layer

(d) performs all functions of the physical layer

5.2 Software in PC

The software in the PC, written in Turbo Pascal, performs some functions of both the application layer and the transport layer. One important feature of this software is the friendly, easy-to-use, menu driven, human interface. The techniques of multiple windows and pop-up windows are employed to achieved effective interaction with the user. Its software functions are performed through its main routine and 17 subroutines (Figure 5.1).

5.2.1 Main Routine of Software in PC

The major function of the main routine is to provide the first level user interface. It begins with the initialization of the COM 1 serial port and the presetting of all the data used by the screen window subroutines. Next, it opens three windows. The first window displays the current disk drive and path name, the second window shows the directory listing of files in the current path, and the third window displays the command menu of network applications. Seventeen network applications are presented in the command menu. The user selects a desired application using two arrow keys (up and down) and initiates the execution of the corresponding subroutine by pressing the return key. The "Esc" key allows the user to leave the main routine and return to DOS.

5.2.2 Subroutines of Software in PC

The seventeen subroutines executable through the command menu are

1. Procedure mail_directory,
2. Procedure read_mail,
3. Procedure purge_mail,
4. Procedure transmit_mail,
5. Procedure remote_task,
6. Procedure local_task,
7. Procedure network_status,
8. Procedure terminal_mode,
9. Procedure monitor_bus,
10. Procedure run_program,
11. Procedure make_subdirectory,
12. Procedure change_subdirectory,
13. Procedure change_log_drive,
14. Procedure delete_file,
15. Procedure rename_file,
16. Procedure list_file, and
17. Procedure show_directory.

Procedure mail_directory

This subroutine is executed when the user selects the item "Mail directory" from the command menu. It displays a directory listing of the electronic mail box in the NIU, indicating the number of messages received, the senders of individual messages, and the memory locations where they are stored in the NIU.

In order to obtain a directory listing from the NIU, the PC issues a local task command packet, {00!2A.}, to the

NIU via the serial link. A software subroutine of the task library gathers a complete directory listing and sends it back to the PC via the same serial link. The software subroutine which gathers the directory is a member of the task library, with the task number of 2A.

Procedure read_mail

This subroutine is executed through the item "Read mail" of the command menu. It provides random access of the messages in the electronic mail box. The user can access any message by specifying the message number, and may elect to browse it on PC's screen or have it recorded in the disk memory at the same time. Since the electronic mail box is in the NIU, the PC obtains the desired message, e.g. message 1, by issuing the local task command packet, {00!26.1}, via the RS-232c link. A task in the task library, with the task number of 26, performs the transmission of message 1 to the PC via the serial link.

Procedure purge_mail

This subroutine can be accessed by selecting the item "Purge mail" from the command menu. In an effort to prevent accidental erasure of messages in the electronic mail box, it warns the user by printing the question "Are you sure?" on the screen. This subroutine purges all the messages in the electronic mail box by issuing a local task command packet, {00!2C.}, to the NIU. A software subroutine of the task library, with the task number of 2C, performs the actual purging.

Procedure transmit_mail

This subroutine is executed when the user selects the item "Transmit mail" from the command menu. To send a message, the user begins by specifying the destination address. Next, a pop up menu with four options is presented.

1. Send message to mail box of recipient.
2. Send message to mail box and printer of recipient.
3. Send message to mail box and local serial port of recipient.
4. Send message to mail box and PC of recipient.

After choosing one of the options above, the user will be offered two more alternatives:

1. Send a short message which can be created by typing it on the PC screen (up to 700 Bytes).
2. Send a message which has previously been created and stored as a disk file (up to 8 KByte).

To transmit a message, the PC sends a transport packet (TP) to the local NIU. For the purpose of illustration, the network address of the local node is assumed to be 10 or 0AH in Hex. To send a message to the destination address of 11 (0BH in Hex), four types of transport packet are possible:

1. [OBOA{}message,EOF] for sending a message to the mail box of recipient.
2. [OBOA{:27.}message,EOF] for sending a message to the mail box and the printer of recipient.
3. [OBOA{:28.}message,EOF] for sending a message to the mail box and the local serial port of recipient.
4. [OBOA{:26.}message,EOF] for sending a message to the mail box and PC of recipient.

The term "EOF" represents the ASCII end-of-file character, or 1AH in Hexadecimal.

Procedure remote_task

This subroutine is executed through the item "Remote tasks" of the command menu. It displays a pop-up menu which offers 17 possible remote tasks. To schedule a remote task, the PC issues a transport packet (TP) to the local NIU. For the purpose of illustration, the network address of the local node is assumed to be 10 or 0AH in Hex, and the destination address to be 11 (0BH in Hex). If the menu item "Reset remote node" is chosen then the transport packet will be

[OBOA{!00.}EOF].

Procedure local_task

This subroutine is executed through the item "Local tasks" of the command menu. It displays a pop-up menu which offers 22 possible local tasks. To schedule a local task, the PC issues a local task command packet (LTCP) to the local NIU. For example, if the local task chosen is "Reset local node" then the LTCP will be

{00!00.}.

Procedure network_status

This subroutine is executed when the user selects the item "Network status" from the command menu. It displays a listing of all the network nodes that are currently active. This listing can be generated by the execution a special task in the NIU. In order to obtain a network status listing from the NIU, the PC issues a local task command packet, {00:31.}, to the NIU via the serial link. A software subroutine of the task library, task 31, gathers a complete network status listing and sends it back to the PC via the same serial link.

Procedure terminal_mode

This subroutine is executed through the item "Terminal mode" of the command menu. Its purpose is to operate the PC as a dumb terminal for asynchronous communication with the NIU at 9600 baud.

Procedure monitor_bus

This subroutine is executed when the user selects the item "Bus monitoring" from the command menu. It schedules the execution of a special local task that converts the NIU into a passive dedicated system capable of capturing all the data traffic passing through the network bus. The local task command packet for this special task is {00!29.}. If necessary, the user may record the captured data in a disk file. Troubleshooting network problems is made possible by analyzing the captured data. It serves an important role in the management and maintenance of the local area network.

Procedure run_program

This subroutine is executed when the user selects the item "Run program" from the command menu. It utilizes the DOS EXEC call, function 4B, to run another program (external program) in its PC; and regains the control of the PC when the external program terminates. The user needs not terminate the current LAN software in the PC in order to run another program. For example, the user may run a word processor (external program) to edit a large text file and return to the main routine to transmit the edited file to another user.

Procedure make_subdirectory

This subroutine is executed through the item "Make Directory" of the command menu. It utilizes the built-in Turbo Pascal procedure, MkDir(parameter), to create a new sub-directory as specified by the path name supplied in the parameter.

Procedure change_subdirectory

This subroutine is executed when the user selects the item "Change directory" from the command menu. The built-in Turbo Pascal procedure, ChDir(parameter), is used to change the current subdirectory to the path name specified by the parameter.

Procedure change_log_drive

This subroutine can be executed via the item "Change disk drive" of command menu. It utilizes the built-in Turbo Pascal procedure, ChDir(parameter), to change the current disk drive to that specified by the parameter.

Procedure delete_file

This subroutine is executed through the item "Delete file" of the command menu. Its function is to delete any disk file specified by the user. It performs the delete-file function with the built-in Turbo Pascal routine, DELETE(filename).

Procedure rename_file

This subroutine is executed when the user selects the item "Rename file" from the command menu. The user may use it to rename any disk file. A built-in procedure of Turbo Pascal, RENAME(old_name,new_name), performs the renaming of a disk file.

Procedure list_file

This subroutine can be executed via the item "List file" of the command menu. Its function is to display the contents of any disk file selected by the user. It uses several file handling routines of Turbo Pascal to perform the file listing.

Procedure show_directory

This subroutine is accessed through the item "Show file info" of the command menu. Its function is to display the information of a disk file, which includes the file size in bytes, the time and date when the file is created or updated, and its attributes. File information is obtained by invoking DOS function calls, 4E and 4F.

5.3 Software in NIU

The software in NIU is written in the low level assembly language of the 8031 microcontroller. It performs some functions of the application layer and transport layer as well as all the functions of the data link layer and physical layer. The software in NIU will be discussed in terms of memory mapping and software functions implemented.

5.3.1 Memory Mapping

The memory mapping of the NIU is dictated by the architecture of the microcontroller chip (8031). Two types of memory addressing, the internal memory (on-chip) and the external memory (data and program), are provided.

Internal Memory

The microcontroller chip (8031) has 128 bytes of on-chip internal data memory (Figure 5.2). They are necessary for the execution of time critical functions and are assigned as follows:

00H to 07H --> Register bank 0 for queued task.
08H to 0FH --> Register bank 1 for immediate task.
10H to 17H --> Register bank 2 for timer interrupt service.
18H to 1FH --> Register bank 3 for main routine.
20H to 2FH --> Dedicated registers and flags.
30H to 47H --> System stack.
48H to 57H --> Buffer for local task command packet (LTCP).
58H to 67H --> Buffer for network task command packet (NTCP).
68H to 7FH --> Task queue.

External Memory

All external memory and I/O on the SIBEC II circuit board have been assigned to a common memory space. The external addressing space provided by the microcontroller system is 48 KByte. Eight (8) KByte of memory are allocated to PROM for program memory and 24 KByte of memory to RAM for data and buffers. The serial communication controller (SCC) and programmable peripheral interface (PPI) are also mapped into the external memory space. External memory space is allocated as follows (Figure 5.3):

0000-1FFF : PROM, program memory for software in NIU.
2000-27FF : RAM, dedicated registers.
2800-2FFF : RAM, buffer for TP generated in NIU.
3000-4FFF : RAM, buffer for TP generated in PC.
5000-7FFF : RAM, packets received from other nodes.
8000-BFFF : Unused, available for future expansion.
C000-C003 : Serial communication controller (82530).
F000-F003 : Programmable peripheral interface (8255).

The data addressing space can be expanded to BFFFH to provide another 16 KByte of memory locations. This requires some minor modifications in the hardware of the NIU. The 8031 microcontroller chip also supports separate external memory spaces for program and data, each up to 64 KByte. With some hardware modifications, the microcontroller system may utilize separate program and data addressing spaces to get more speed and data memory.

5.3.2 Software Functions

The software in the NIU is designed to provide a pseudo multitasking capability with the use of multiple interrupt service routines. The NIU appears to handle several functions at the same time. For example, it can simultaneously receive data from both the PC and the network bus while printing hardcopies via its Centronics port.

The software operation can be divided into four main functional modules: the main routine loop, the timer routine, the routine to receive data from network bus, and the routine to receive data from PC (Figure 5.4). The timer routine interrupts the main routine loop at a constant interval of 10 msec to maintain an accurate real-time clock. The remaining two functional modules are interrupt routines which occur randomly within a small time window in the main routine loop.

5.3.3 Main Routine Loop

The NIU settles into the main routine loop after system initialization (Figure 5.5). The main routine loop is interrupted every 10 msec by the timer routine. Furthermore, a small time window is opened in the main routine loop to accept two more random interrupts, one for "routine to receive data from bus" and the other for "routine to receive data from PC."

An internal flag, SM2, of the microcontroller chip is preset to "1" after system initialization to ensure that the main routine will not be unnecessarily interrupted by data on the network bus. It will, however, permit interrupts by destination addresses of data packets.

The main routine loop begins with the supervision of background printing activities. It skips to the next function if the printer is busy, otherwise it examines the print queue to decide whether the idling printer should get busy again. If the print queue is not empty, a task command (instruction) to initiate the background printing of the next file is placed in the task queue. An empty print queue simply means that no file awaits printing and that the printer may remain idle.

Next, it assembles a media access packet (MAP) from the data supplied by the transport packet (TP). The transmission of the assembled MAP to one or more intended nodes is handled by the "transmit MAP" subroutine.

The next function of the main routine loop executes a queued task, if any exists.

The main routine repeats itself in an infinite loop until a system reset occurs.

Transmit MAP Subroutine

The "transmit MAP" subroutine is part of the main routine loop. It performs the transmission of the MAP to one or more nodes. As shown in the flow chart (Figure 5.6), it remains in the "ready mode" as long as the network bus is busy, but immediately initiates the transmission of a MAP when the network bus is free. If a collision is detected while transmitting, it terminates the current transmission and schedules a retransmission after a backoff period. The backoff period is proportional to its node address. It repeats the retransmission process whenever a collision is detected.

The activity after transmitting a complete MAP depends on the type of data link set up by the destination address of the MAP. It exits to the main routine loop if an unacknowledged connectionless link has been set up with a universal address. Otherwise, it waits for an acknowledgement if a connection-oriented link has been set up using a unique destination address. The connection-oriented link demands the return of an acknowledgement from the receiving node within a specified time (timeout). A failure will cause a retransmission to be scheduled. The mission to transmit a packet is abandoned after three unsuccessful attempts.

5.3.4 Routine to Receive Data from Network Bus

This is an interrupt service routine initiated by the reception of data from the network bus (Figure 5.7). The reception of data and the subsequent interrupt are handled by the serial port of the 8031 microcontroller chip. After system reset, the physical layer operates in "idle mode" since the SM2 flag of the microcontroller chip has been set to 1. In this configuration, it only permits an interrupt caused by the arrival of destination addressing characters while ignoring all other passing data.

The physical layer switches to "input mode" with the SM2 flag cleared to 0 when it has received two identical destination addresses which either match its own address or the universal address (00). The clearing of the SM2 flag (SM2=0) enables the reception of all the data that follow the destination addresses.

Upon receiving a complete packet, it examines the checksum to detect any possible data error. A damaged data packet, i.e. one with data error, will be abandoned and no acknowledgement will be returned to the transmitting node. The response to an error free packet, on the other hand, depends on the type of data link that has been set up by the destination address. No acknowledgement is required for an unacknowledged connectionless link set up by a packet with a universal address. In the case of a connection-oriented link, i.e. a data link set up by a packet with a unique destination address, an error free packet is immediately acknowledged.

The next function of this interrupt service routine is the processing of the network task command included in the packet just received.

The physical layer is switched to "idle mode" by setting SM2 to 1 before exiting to the main routine loop.

5.3.5 Routine to Receive Data from the PC

This is an interrupt service routine initiated by the reception of data from the PC (Figure 5.8). The serial communication controller (82530) is responsible for the reception of data and the subsequent output of interrupt signals to the external interrupt pin (EX1) of the 8031 microcontroller chip.

Two types of packets are expected from the PC, i.e. the transport packet (TP) and the local task command packet

(LTCP). They are handled in different manners. Upon receiving a complete transport packet from the PC, it merely evaluates the checksum before exiting to the main routine loop. Whereas, a local task command packet will be immediately processed and executed according to the directives specified by the task command. The transport packet will be further processed in the main routine before it is transmitted to the intended destination in the form of a MAP.

5.3.6 Timer Routine

This is a high priority interrupt service routine which breaks into operation every 10 msec (Figure 5.9). Its constant periodic interrupts are initiated by the internal timer of the 8031 microcontroller chip. Because of its "high priority" status, the timer may interrupt a low priority interrupt service routine to maintain a constant interval between interrupts. Therefore, the "high priority" status of the timer interrupt is necessary for the maintenance of an accurate real-time clock. The resident microcontroller system, the enhanced TASKMASTER, relies on the accuracy of the real-time clock to terminate a task whose allocated execution time has expired.

A minor function of this routine is the indication of data collision in the transmitting node by blinking the red LED.

Its last function, the execution of background printing, is carried out by sending one file character to the printer before leaving the timer routine. Since the timer interrupt occurs every 10 msec, the file data are, therefore, transmitted to the printer at the rate of 100 characters per second.

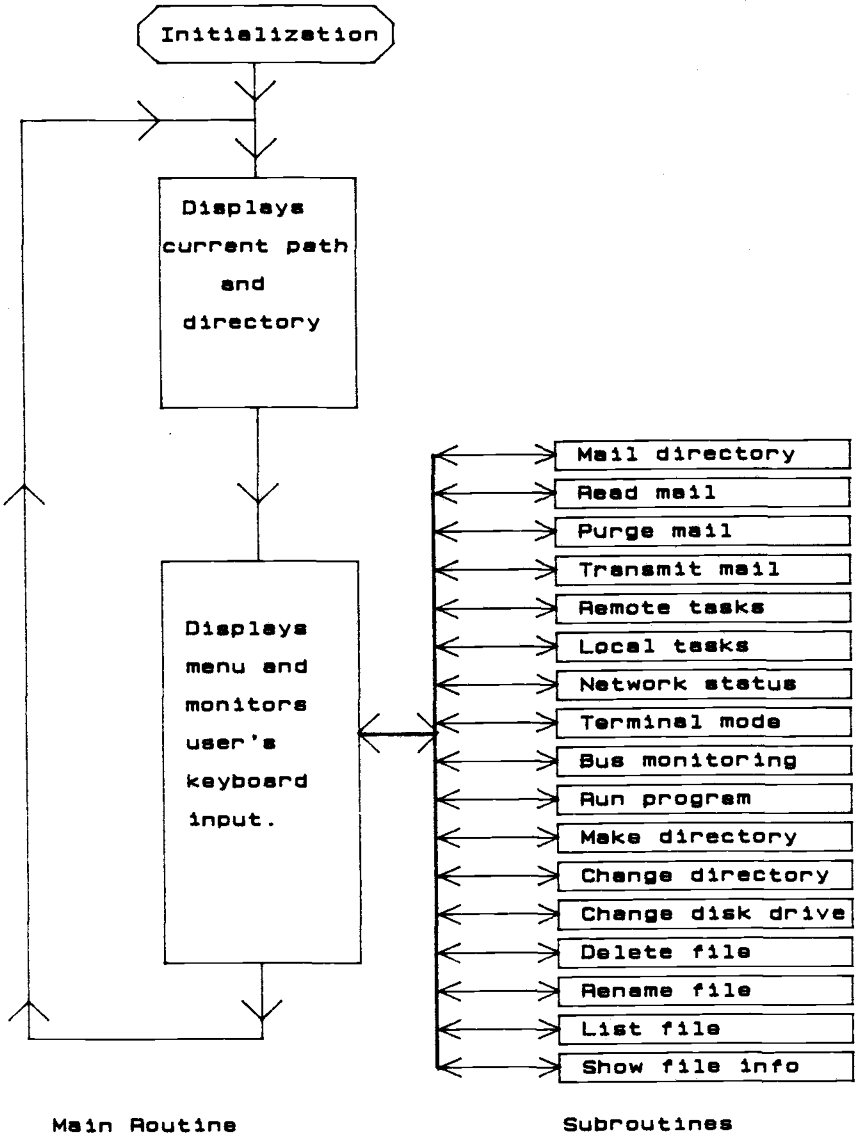

Figure 5.1 Conceptual Configuration of Software in PC

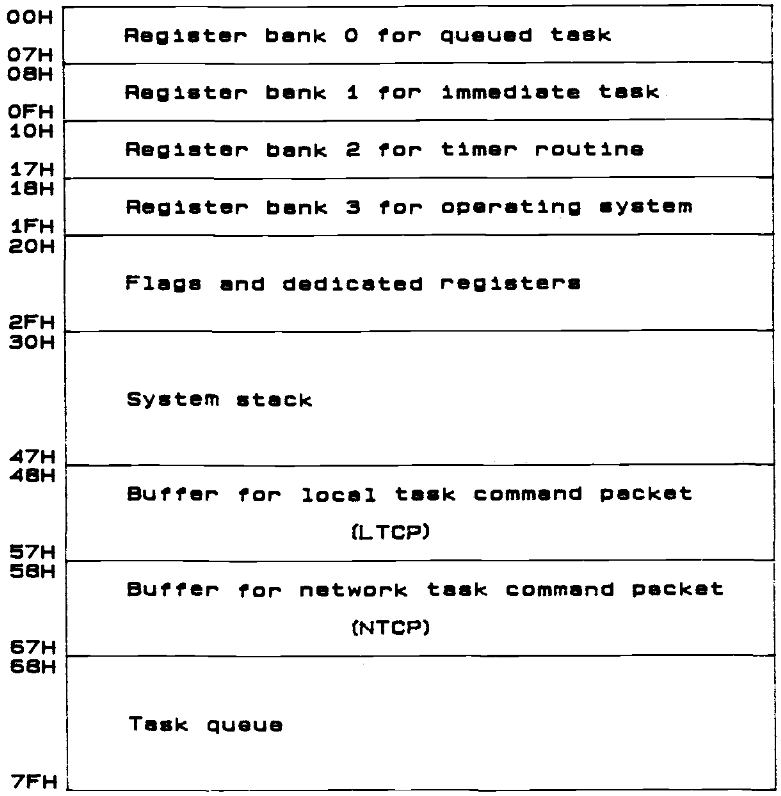

Figure 5.2 On-chip Internal Memory Map

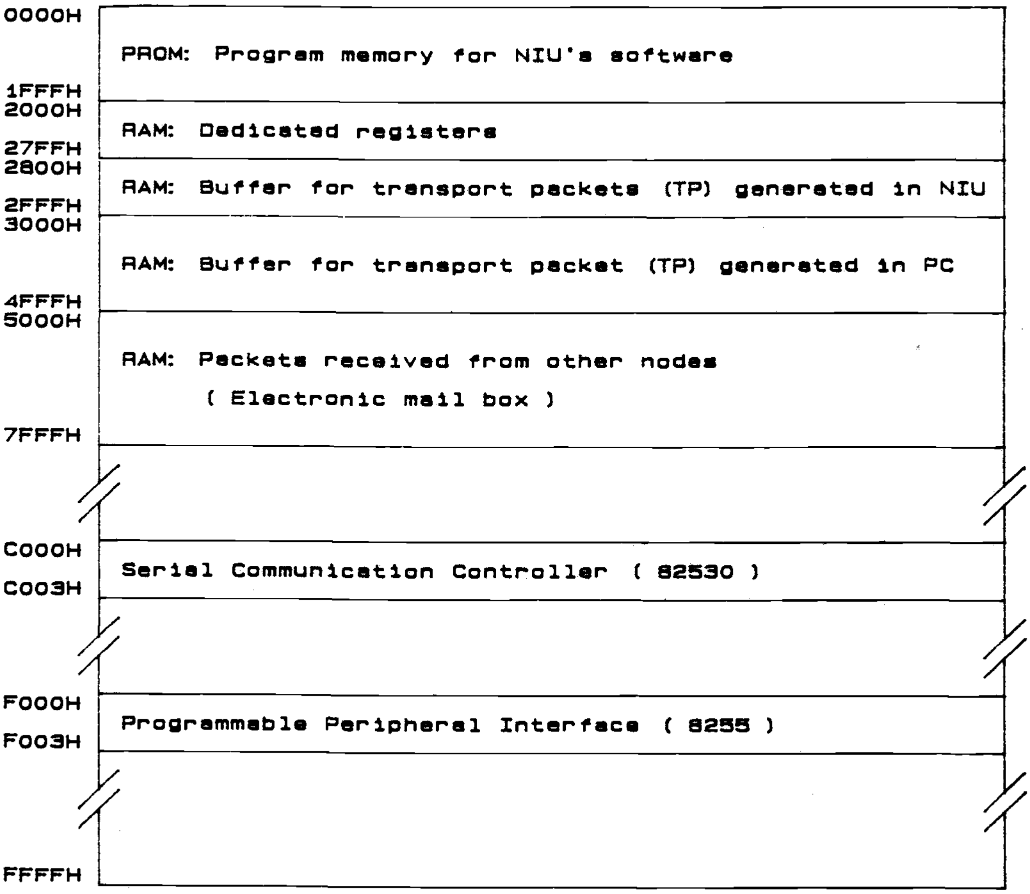

Figure 5.3 Addressing Space of External Memory

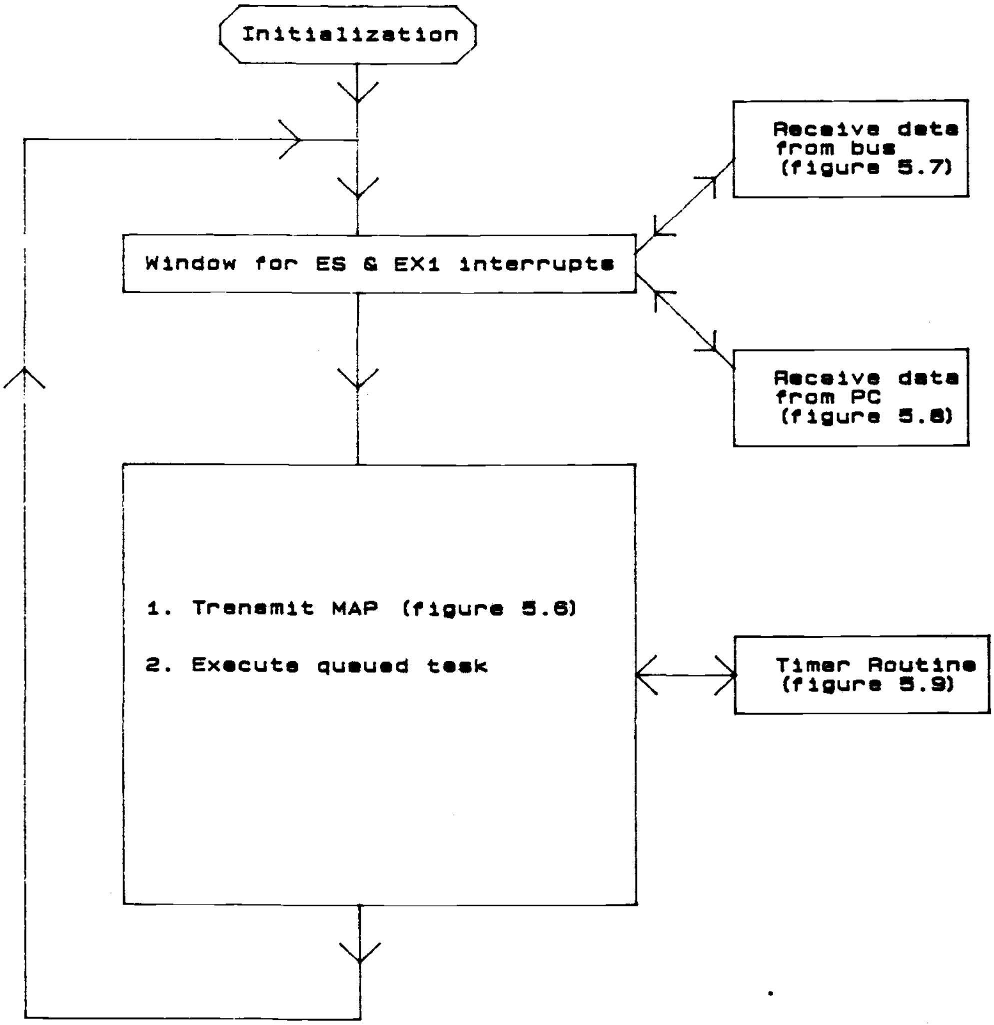

Figure 5.4 Conceptual Configuration of Software in NIU

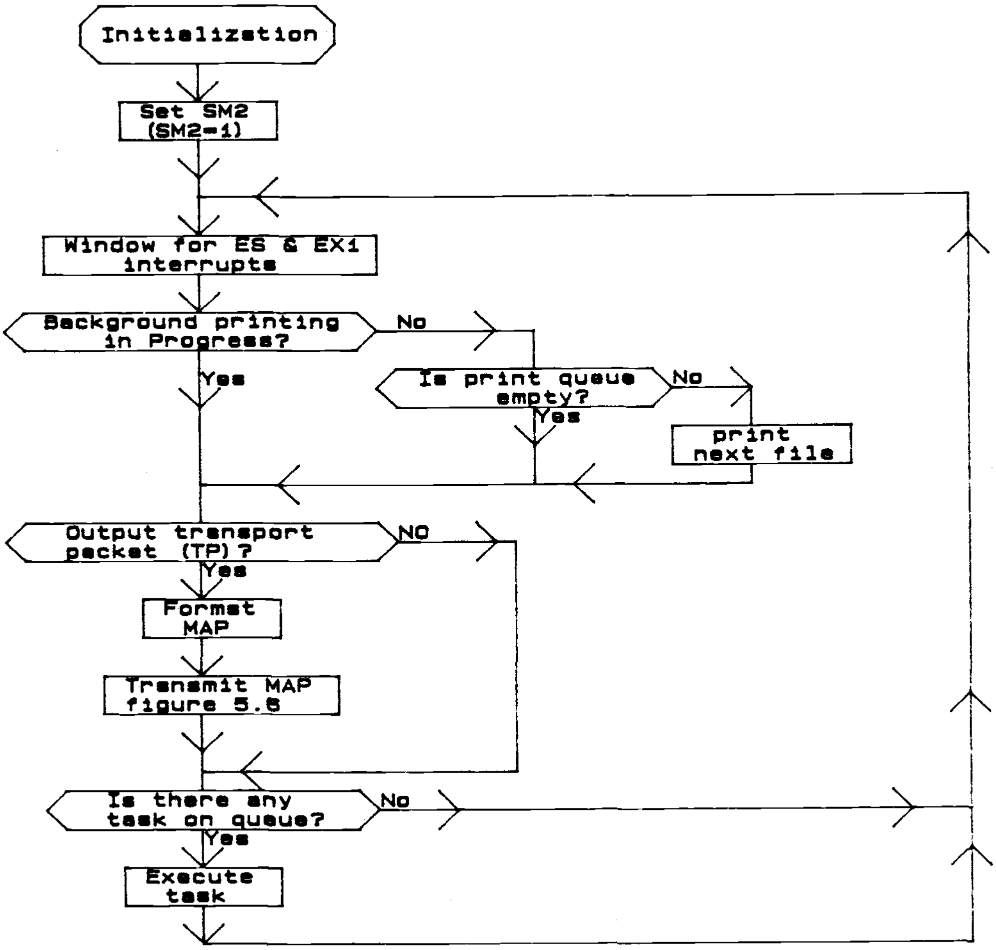

Figure 5.5 Main Routine Loop

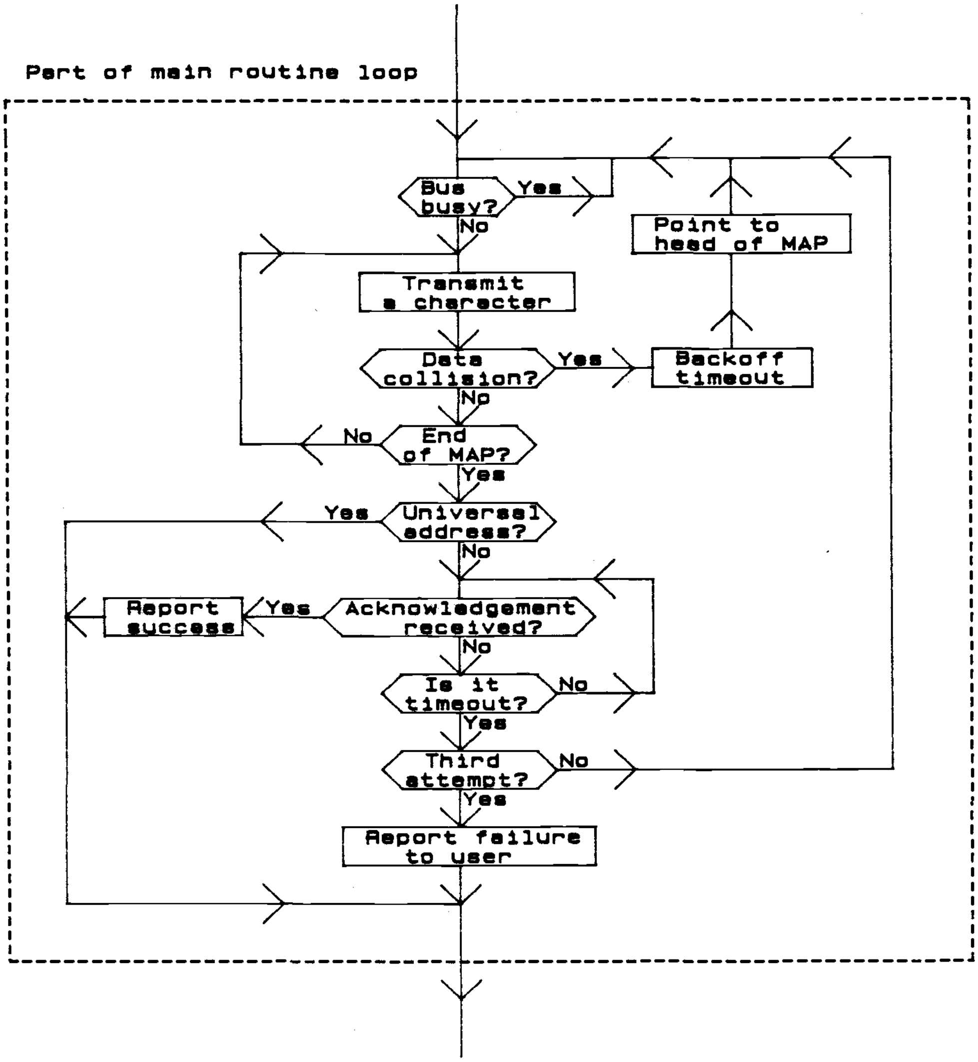

Figure 5.6 Transmit MAP Subroutine

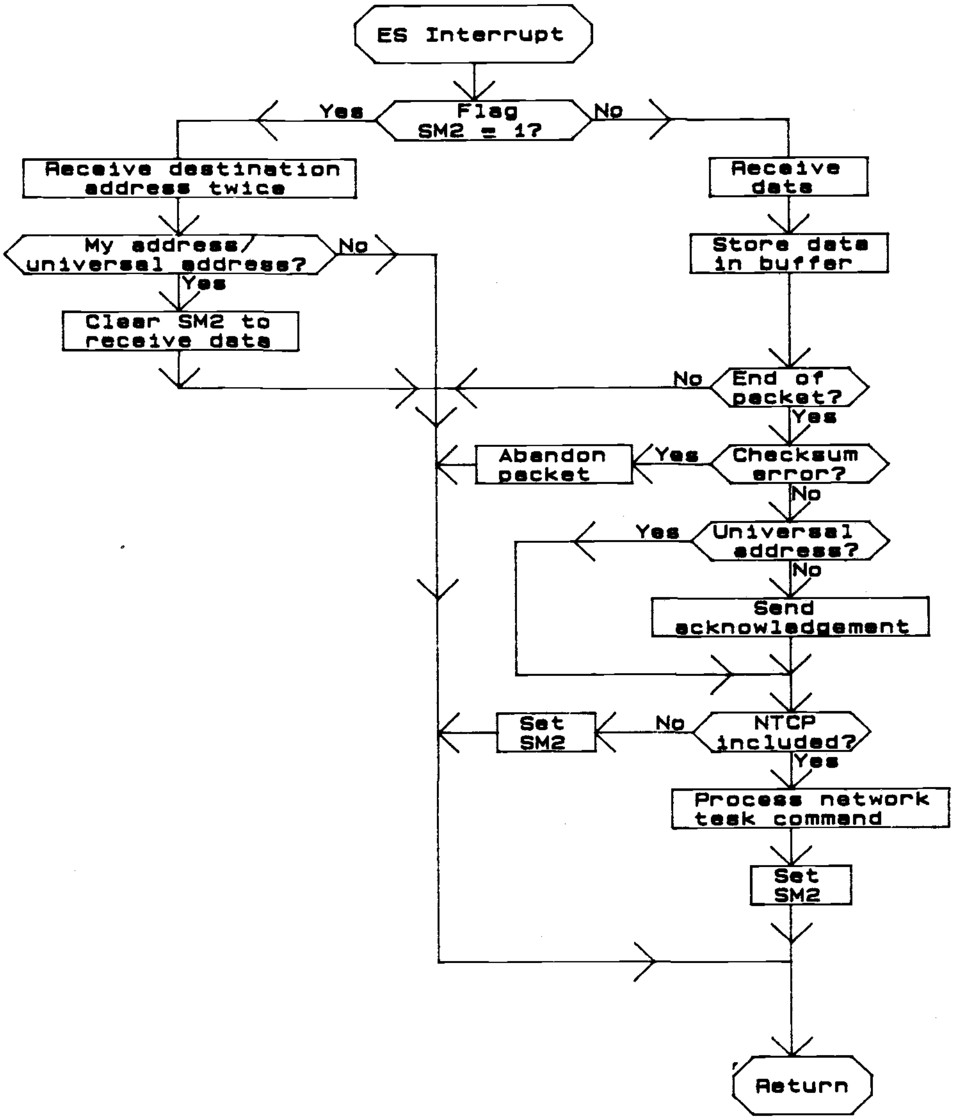

Figure 5.7 Routine to Receive Data from Network Bus

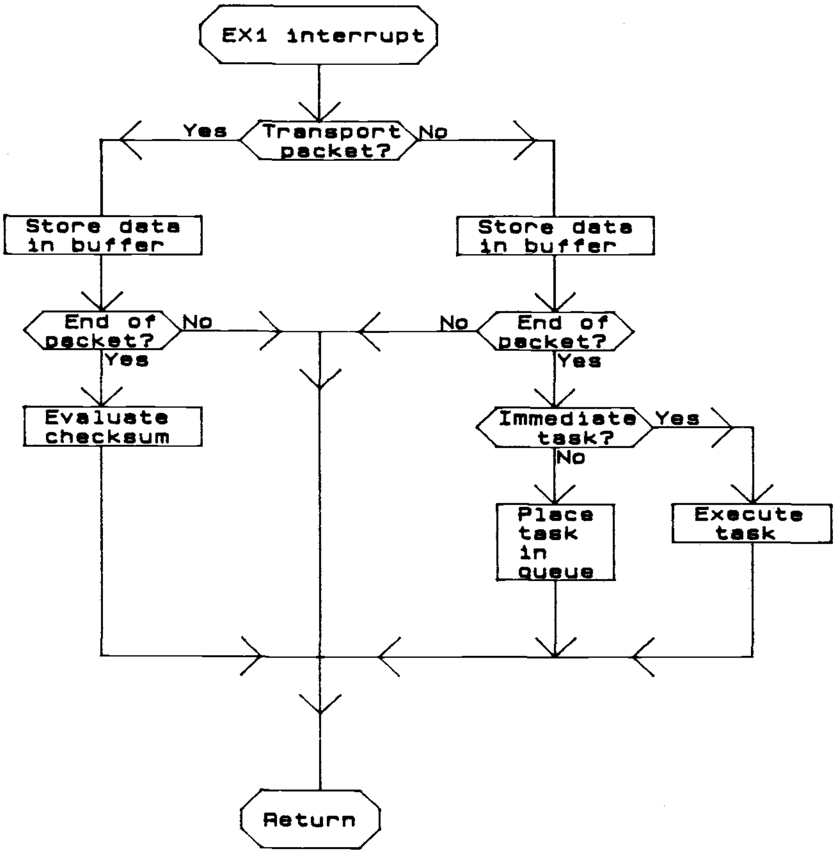

Figure 5.8 Routine to Receive Data from PC

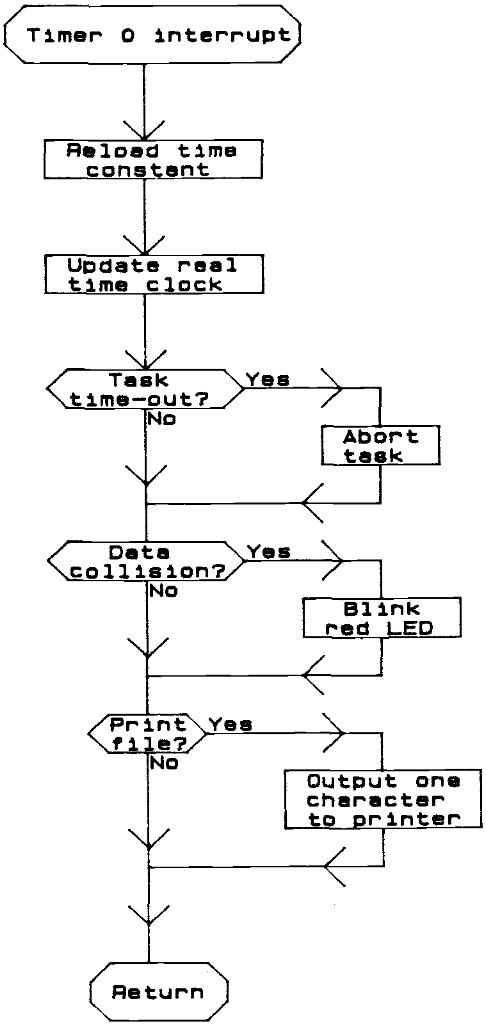

Figure 5.9 Timer Routine

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS

6.1 Conclusions

Every feature of COLAN IV has been tested with numerous experiments performed with four network nodes. Ideally, COLAN IV should be pushed to its functional limits by testing it with the maximum load of 256 nodes and the limiting cable length of 4000 ft. But due to the lack of available hardware components only four nodes have been implemented for investigation. However, satisfactory results from those experiments performed with four nodes confirm the viability of COLAN IV as a dual purpose LAN.

Numerous investigative experiments have been conducted on COLAN IV for three different roles:

1. As a communications network.
2. As a distributed control system.
3. As a combination of both communications network and distributed control system.

When used as a communications network, it consistently exhibited a reliable performance, and demonstrated the successful implementation of the CSMA/CD media access technique. The network bus monitoring function was used during these experiments to record the data activities on the bus. The analysis of these data activities revealed that the process of bus contention resolution after a data collision has been properly executed by the participating nodes on the LAN. Unfortunately, it intermittently failed during the execution of the "network status" function, a communication application which generates a listing of active nodes on the bus. An obscure software bug is suspected. Although under investigation, this design flaw has not been eradicated.

When used as a distributed control system, the results confirm that COLAN IV can support a wide variety of control applications. In these experiments, coordinated actions of a control system were achieved by scheduling a series of synchronized tasks on several NIUs which served as distributed controllers. A PC, which served as the system scheduler, managed the coordinated actions of these multiple microcontrollers by the use of synchronizing commands.

Outcomes of investigative experiments also confirm that COLAN IV can simultaneously support a distributed control system while operating as a communications network.

Every effort has been made to provide a transparent network operation to users by offering a friendly, easy-to-use, human interface via the software in the PC.

The three features of the COLAN IV, i.e., the simplicity of bus topology, the "fair" access and distributed control of CSMA/CD technique, and the provision of the network bus monitoring function, make it a highly reliable and maintainable system.

COLAN IV supports conventional applications of a LAN such as data communications, electronic mail, and resource sharing. Its added capability to support several distributed control systems within the same network greatly enhances its usefulness.

6.2 Suggestions for Future Developments

Although the current configuration of COLAN IV satisfies the level of expectation for this thesis, it has ample room for improvements. The following suggestions are recommended for the future development of COLAN IV.

1. Performance of COLAN IV can be greatly improved by replacing the current 8051 microcontroller of the NIU with an 8044 microcontroller chip. This chip achieves a much better performance by integrating an 8051 microcontroller and an intelligent serial coprocessor unit together in one chip. In addition, the 8044 microcontroller supports data rates up to 2.4 Mbps, surpassing the maximum data rate of 100 Kbps allowed for a 4000 ft cable. The RS-485 bus standard prohibits data rate higher than 100 Kbps while operating with the maximum permissible bus length of 4000 feet.

2. The Cyclic Redundancy Codes (CRC) offer a better method of data error detection. COLAN IV uses a simple checksum

technique for error detection. Transmission error control will be more efficient if the current checksum technique is replaced by CRC.

3. Performance of the COLAN IV can be further improved by operating at the data rate of 100 Kbps. The absolute limit in the length of twisted-pair cable that can be supported by the RS-485 bus standard is 4000 ft. In order to take advantage of the maximum cable length, the data rate must be limited to 100 Kbps. The data rate may be increased at the expense of cable length. At 10 Mbps the permissible cable length is drastically reduced to a mere 50 ft.

4. To improve channel efficiency of the network bus, a packet switching technique may be adopted to replace the current message switching technique. At present, a node wishing to transfer a long file transmits the entire file in a single packet, i.e. a complete message. Data transactions are carried out among nodes by means of complete messages. The packet switching technique breaks a message into small segments. With message switching technique, a node must retransmit the entire file, or message, if a few characters have been damaged by noise. This leads to inefficient utilization of the bus. A packet switching technique, on the other hand, needs only retransmit a small packet instead of the entire file.

5. Link-list file management may be utilized to provide a better electronic mail box in the NIU. The link-list technique allows random access and deletion of files. The current file management of COLAN IV does not support random deletion of file. It, however, allows random access of files.

BIBLIOGRAPHY

CORS 86 Corso, D. D. and Kirrman, H. and Nicoud J. D. Microcomputer Buses and Links, Academic Press Inc. (London), 1986.

CROW 86 Crowder, R. S., "MAP, Proway, and IEEE 802: A Marriage of Standards for Automation," INFOCOM '86, IEEE Computer Society, 1986, pp.337-342.

EUM 87 Eum, D. COLAN III, A Control-Oriented LAN Using CSMA/CD Protocol, Unpublished Master's thesis, Oregon State University, Corvallis, Oregon, September 1987.

FARO 86 Farowich, S. T., "Communication in the Technical Office," Spectrum, April 1986, IEEE Inc.

HAMM 86 Hammond, L. J. and O'Reilly, J. P. Performance Analysis of Local Computer Networks, Addison-Wesley Publishing Company, 1986.

HERZ 87 Herzog, J. H., "A Design Methodology for Distributed Microprocessors in Real-time Control applications." Paper presented at the Second International Conference on Computers and Applications, Beijing, People Republic of China, June 1987.

IEEE 85a IEEE Std 802.3, Local Area Network Standard-Carrier Sense Multiple Access with Collision Detection, IEEE Inc., 1985.

IEEE 85b IEEE Std 802.4, Local Area Network Standard-Token Passing Bus Access Method, IEEE Inc., 1985.

IEEE 85c IEEE Std 802.5, Local Area Network Standard-Token Ring Bus Access Method, IEEE Inc., 1985.

IEEE 85d IEEE Std 802.2, Local Area Network Standard-Logical Link Control, IEEE Inc., 1985.

INTE 83 Intel Corporation. Microcontroller Handbook, 1983.

JUAN 86 Juang, J. Y., and Wah, B. W., "Channel Allocation in Multiple Contention Networks," INFOCOM'86, IEEE Computer Society, 1986, pp.189-196.

KAO 87 KAO, S., Design of COLAN II, A Control Oriented Local Area Network, Unpublished Master's thesis, Oregon State University, Corvallis, Oregon, September 1987.

LINE 86 Linebarger, R. N. and Dogget, D. D., "Resource Sharing Local Networks," 11th Conference on Local Computer Networks, IEEE Computer Society Press, Washington D.C., 1986.

MALH 86 Malhotra, M. K., and Omidyar, C. G., "Trends and Issues in Local Area Networks," INFOCOM'86, IEEE Computer Society, 1986, pp.123-129.

MAYN 86 Mayne, A. J., Linked Local Area Networks, John Wiley and Sons Inc., New York, 1986.

MEDI 86 Meditch, J. S., and Yin, D., "Performance Analysis of Virtual Time CSMA," INFOCOM 86', IEEE Computer Society, 1986, pp.242-251.

MINN 86 Minnich, N. M., "A Packet Capture System for LAN Software Development," 11th Conference on Local Computer Networks, IEEE Computer Society Press, Washington D.C., 1986.

PICK 86 Pickholtz, R. L., Local Area & Multiple Access Networks, Computer Science Press, Inc., 1986.

STAL 84 Stalling, William., Local Networks, Macmillan Publishing Company, 1984.

TANE 81 Tannenbaum, A. S., Computer Networks, Englewood Cliffs, NJ, Prentice Hal Inc., 1981.

TASA 86 Tasaka, S., Performance Analysis of Multiple Access Protocols, The MIT Press, Cambridge, Massachusetts, 1986.

ZHEN 86 Zheng, Y., A Simple Local Area Network, COLAN (Control Oriented Local Area Network), Unpublished Master's thesis, Oregon State University, Corvallis, Oregon, December 1986.

Appendix

APPENDIX A

SETTING-UP PROCEDURES AND OPERATIONAL GUIDE

The materials in this appendix describe the procedures for setting up COLAN IV and provide an operational guide to first-time users.

A.1 Setting-Up Procedures

Each node of COLAN IV consists of two hardware components, the IBM PC and the NIU. A shared network bus links several nodes into a LAN. A functional COLAN IV can be set up by the following steps.

1. Complete the RS-232c serial link within each node by connecting the COM 1 serial port of the IBM PC to the RS-232c port of corresponding NIU (Figure A.1).

2. Complete the shared network bus by linking adjacent NIUs with standard 4-conductor telephone wires. The dual-socket telephone connector of each node provides the mechanical coupling to the linking telephone wires (Figure A.1). The result should be a bus with no circular paths.

3. Power up all NIUs when the network bus connection has

been completed. The operating system of each NIU automatically completes the system reset routine after power-up, and prepares to execute network applications.

4. Run the high level software of COLAN IV in each IBM PC.

As soon as the "command menu" appears on the screen of every PC, the user may begin the execution user applications. The friendly, easy-to-use, menu-driven, human interface software will guide the user through the necessary operating steps.

A.2 Operational Guide

All user applications can be executed through the "command menu" presented on the screen of the IBM PC. Seventeen applications are offered in the command menu. The user can select a menu item by using two arrow keys (up and down) and confirm the selection by pressing the "enter" key. The "ESC" key allows the user to leave the command menu and return control of the PC to DOS. When a command is selected, a new menu will be presented to guide the user.

The PC displays the node address at the top right hand corner of the screen. If the local NIU is down for some reason, the PC will display a blinking message "node not ready" which overwrites the node address.

The following sections provide the operational guide to the execution of seventeen applications offered by the command menu.

1. Mail Directory

When the command "Mail directory" is chosen, the PC displays a directory listing of the electronic mail box in the NIU. The user can return to the command menu by pressing the "ESC" key.

2. Read Mail

When the command "Read mail" is chosen, the user can randomly access any message in the mail box by specifying the message number. The user may elect to browse the message on the screen or have it written to a disk at the same time. This read process does not erase the message in the mail box. The "ESC" key allows the user to return to the command menu.

3. Purge Mail

Through this command the user may purge all the messages in the mail box. Unfortunately, the software does not support random deletion of a message. To prevent accidental erasure of all the messages in the electronic mail box, it warns the user by printing the question "Are you sure?" on the screen.

4. Transmit Mail

Through this command the user can send a message to another node on the LAN. To send a message, the user begins by specifying the destination address. Next, a pop up menu with four options is presented to the user.

1. Send message to mail box of recipient.
2. Send message to mail box and printer of recipient.
3. Send message to mail box and local serial port of recipient.
4. Send message to mail box and PC of recipient.

After choosing one of the options above, the user will be offered two more alternatives:

1. Send a short message which can be created by typing it on the PC screen (up to 700 Bytes).
2. Send a message which has previously been created and stored as a disk file (up to 8 KByte).

After sending a message, the user may use the "ESC" key to return to the command menu.

5. Remote Tasks

When the command "Remote task" is chosen, the PC first requests for the address of the destination node, then presents a pop-up menu which offers 17 possible remote tasks. After selecting a task from the menu, the user has to specify whether it is an immediate task, a queued task, or a synchronized task. The user also has to specify whether it should be requeued after the termination of task execution. Finally, the user has to supply the task parameters if necessary.

6. Local Tasks

The execution of a local task is similar to that of a remote task. It does not require the specification of a destination address because the task execution occurs within the local NIU.

7. Network Status

When the user selects the menu item "Network status" from the command menu, the PC displays a listing of all the network nodes that are currently active.

8. Terminal Mode

This command allows the user to operate the PC as a dumb terminal which communicates with the local NIU at 9600 baud.

9. Bus Monitoring

This command allows the user to convert the local NIU into a passive dedicated system capable of capturing all the data traffic passing through the network bus. If necessary, the user may record the captured data in a disk file.

10. Run Program

This command allows the user to run another program (external program) in its PC; and regains the control of the PC when the external program terminates. The user needs not terminate the high level software of COLAN IV in order to run another program. For example, the user may run a word processor (external program) from the command menu to edit a large text file and then return to the command menu again to transmit the edited file.

11. Make Directory

This command allows the user to create a new subdirectory in the PC by specifying the new path name.

12. Change Directory

This command allows the user to change the current subdirectory of the PC to one specified by the user.

13. Change Disk Drive

This command allows the user to change the current disk drive of the PC, for example, from drive A to drive B.

14. Delete Files

This command allows the user to delete any disk file in the PC.

15. Rename File

This command allows the user to rename any disk file in the PC.

16. List File

This command allows the user to display the contents of any disk file in the PC.

17. Show File Info

This command allows the user to obtain the information of a disk file, which includes the file size in bytes, the time and date when the file is created or updated, and its attributes.

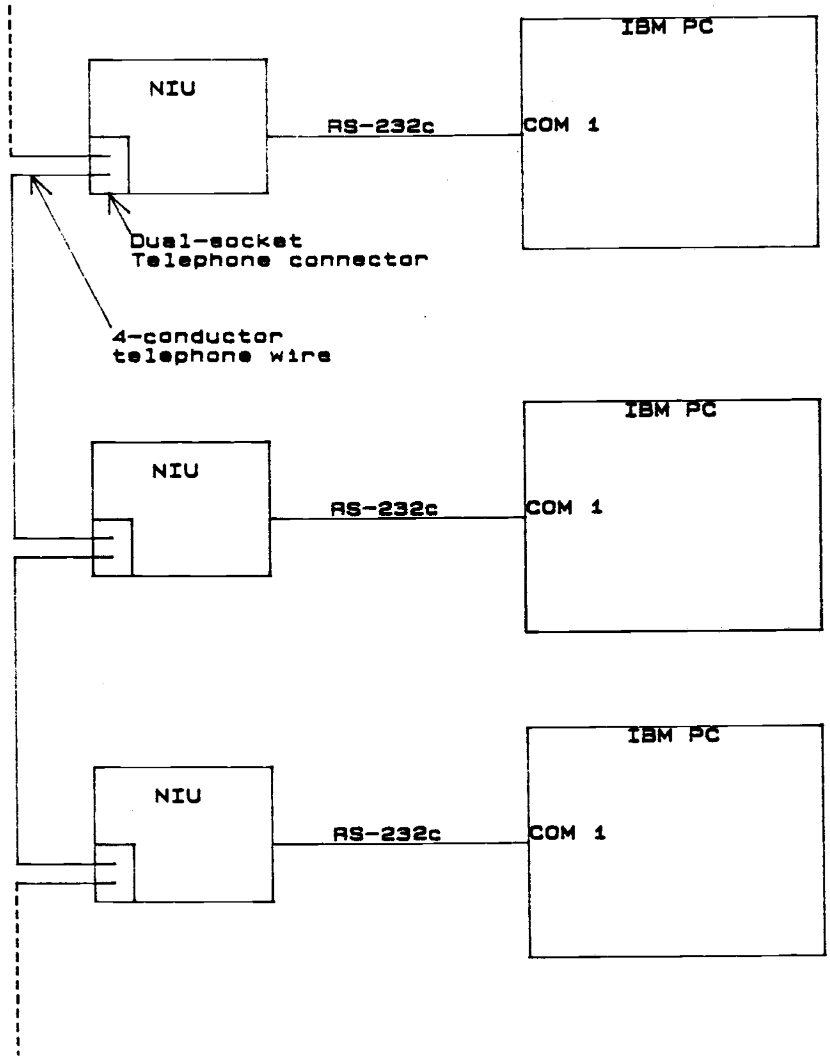

Figure A.1 Hardware Set-up of COLAN IV